

A satellite with solar panels and a dish antenna is in orbit above the Earth's horizon. The Earth shows blue oceans, white clouds, and a small ice cap.

3.

Auflage



Gerhard Lienemann

TCP/IP

Grundlagen und Praxis

Protokolle, Routing, Dienste, Sicherheit

→ Mit Beiträgen von Dirk Larisch

dpunkt.verlag

A close-up view of the Earth's surface from space, showing swirling blue and white cloud patterns over the ocean.



Gerhard Lienemann arbeitete ab 1991 für etwa 12 Jahre als Netzwerkadministrator in einem produzierenden Betrieb, bevor er ins technische Management wechselte. Seitdem war er zunächst zuständig für operative Kommunikationssicherheit und übernahm dann zusätzlich die Betreuung eines europäischen Netzwerkteams. Nur kurze Zeit später erweiterte er seine Verantwortung auf ein globales Netzwerkteam in Asien, Nord- und Südamerika. Seiner Leidenschaft »IT-Kommunikation« ist er auch nach dem Ausscheiden aus dem aktiven Berufsleben treu geblieben.

Copyright und Urheberrechte:

Die durch die dpunkt.verlag GmbH vertriebenen digitalen Inhalte sind urheberrechtlich geschützt. Der Nutzer verpflichtet sich, die Urheberrechte anzuerkennen und einzuhalten. Es

werden keine Urheber-, Nutzungs- und sonstigen Schutzrechte an den Inhalten auf den Nutzer übertragen. Der Nutzer ist nur berechtigt, den abgerufenen Inhalt zu eigenen Zwecken zu nutzen. Er ist nicht berechtigt, den Inhalt im Internet, in Intranets, in Extranets oder sonst wie Dritten zur Verwertung zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und eine gewerbliche Vervielfältigung der Inhalte wird ausdrücklich ausgeschlossen. Der Nutzer darf Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte im abgerufenen Inhalt nicht entfernen.

Gerhard Lienemann

TCP/IP – Grundlagen und Praxis

Protokolle, Routing, Dienste, Sicherheit

3., aktualisierte und überarbeitete Auflage

Mit Beiträgen von Dirk Larisch



dpunkt.verlag

Gerhard Lienemann

feedback@tcip-grundlagen.de

Lektorat: Michael Barabas

Lektoratsassistentz: Julia Griebel

Copy-Editing: Petra Heubach-Erdmann, Düsseldorf

Satz: inpunkt[w]o, Haiger (www.inpunktwo.de)

Herstellung: Stefanie Weidner, Frank Heidt

Umschlaggestaltung: Helmut Kraus, www.exclam.de

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation
in der Deutschen Nationalbibliografie; detaillierte
bibliografische Daten sind im Internet über <http://dnb.d-nb.de>
abrufbar.

ISBN:

Print 978-3-86490-960-3

978-3-96910-957-1

PDFePub978-3-96910-958-8

mobi978-3-96910-959-5

3., aktualisierte und überarbeitete Auflage 2023

Copyright © 2023 dpunkt.verlag GmbH

Wieblinger Weg 17

69123 Heidelberg

Die Voraufgabe dieses Buchs ist 2014 im Heise Verlag
(Hannover) erschienen: Gerhard Lienemann/Dirk Larisch:
TCP/IP – Grundlagen und Praxis, 2., aktualisierte Auflage (ISBN
978-3-944099-02-6).

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus
nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe
verzichten wir zusätzlich auf die Einschweißfolie.



Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen: hallo@dpunkt.de.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort

Vor Ihnen liegt nun das Buch »TCP/IP – Grundlagen und Praxis« in der dritten überarbeiteten Auflage. Für mich als Autor ist dies schon eine überaus bemerkenswerte Situation, da sich das Buch mittlerweile – wenn man die Vorgängerbücher (»TCP/IP – Grundlagen« bzw. »TCP/IP – Praxis«) mit einbezieht – über 25 Jahre im Verkauf befindet und offenbar immer noch den einen oder anderen Interessierten anspricht. Das Thema an sich unterliegt zwar keinen sich stets grundlegend ändernden Technologien (was »eine« Erklärung dafür ist). Allerdings haben sich im TCP/IP-Umfeld über die Jahrzehnte Disziplinen herausgebildet, die ohne TCP/IP und die dazugehörige Protokollfamilie nicht denkbar sind.

Denken wir beispielsweise an das Internet der Dinge (IoT = *Internet of Things*), so beruhen sämtliche heute im Einsatz befindlichen »Smart Devices« immer noch auf Adressierung und Kommunikation nach TCP/IP-Netzwerkprotokollen. Eine programmierbare und über WLAN steuerbare Deckenlampe wird danach genauso »angesprochen« wie ein Router im Netzwerk eines Industrieunternehmens. Das wird sich auch in absehbarer Zukunft voraussichtlich nicht ändern (mal davon abgesehen, dass in der alten IP-Version 4 verfügbare Adressen

knapp geworden sind und daher die neue IP-Version 6 in vielen Fällen bereits »übernommen« wurde).

Dieses und andere aktuelle Themen habe ich nun in dieses Buch aufgenommen und hoffe, damit den heutigen Anforderungen einer ganzheitlichen Betrachtung des Themas Rechnung zu tragen. Sie werden daher auch keine umfänglichen Rückblicke in die Vergangenheit zur TCP/IP-Historie finden, sondern eher Ausblicke in eine mögliche Zukunft.

Damit Sie sich aber gleich zurechtfinden, wenn Sie das Buch in die Hand nehmen, gebe ich Ihnen nun einen Überblick über die Aktualisierungen, die das Buch hoffentlich zu einem sinnvollen Hilfsmittel auch für das TCP/IP der 20er Jahre unseres Jahrhunderts macht:

Die ersten Kapitel werden Ihnen sicher bekannt vorkommen, wenn Sie eine der vorherigen Auflagen des Buches bereits kennen. Es handelt sich hier um Grundlegendes der TCP/IP-Welt. Allerdings habe ich dort, wo es mir sinnvoll erschien, Passagen oder ganze Kapitel weggelassen und neue Passagen bzw. Unterkapitel ergänzt.

In Kapitel 7 (Sicherheit) habe ich beispielsweise im letzten Abschnitt eine völlig andere Denkweise zur Datensicherheit angestoßen, die nicht – wie in der Vergangenheit – primär auf einen Schutz des Perimeters mit Firewalls setzt, sondern einen Sicherheitsschild als Schichtenmodell zur Diskussion stellt, bei dem die Daten im Mittelpunkt stehen.

In Kapitel 8 (Troubleshooting) habe ich exemplarisch das bekannte Analysetool »WireShark« vorgestellt.

Dem Buch habe ich dieses Mal auch einen Anhang beigefügt, in dem die Themen »Das neue TCP/IP-Umfeld« und »TCP/IP-Konfiguration« angesprochen werden. Während das neue TCP/IP-Umfeld eine kurze Reise in das »Internet der Dinge« unternimmt und verwandte Themen behandelt, habe ich im zweiten Anhang anhand unterschiedlicher Plattformen bzw. Betriebssysteme gezeigt, wie bzw. wo dort TCP/IP-Parameter konfiguriert werden.

Ich wünsche Ihnen viel Freude mit diesem Buch und bitte um viele kritische Kommentare, die ich dann in eine vielleicht nächste Auflage einfließen lassen kann. Sie erreichen mich per E-Mail unter feedback@tcpip-grundlagen.de.

Gerhard Lienemann

Inhalt

Netzwerke

Netzwerkstandards

OSI als Grundlage

IEEE-Normen

Netzwerkvarianten

Ethernet

Wireless LAN (IEEE 802.11)

Bluetooth

Sonstige Varianten

Netzwerkkomponenten

Repeater

Brücke

Switch

Gateway

Router

TCP/IP – Grundlagen

Wesen eines Protokolls

Low-Layer-Protokolle

Protokolle der Datumsicherungsschicht (Layer 2)

Media Access Control (MAC)

Logical Link Control (LLC)

Service Access Point (SAP)

Subnetwork Access Protocol (SNAP)

Protokolle der Netzwerkschicht (Layer 3)

Internet Protocol (IP)

Internet Control Message Protocol (ICMP)

Address Resolution Protocol (ARP)

Reverse Address Resolution Protocol (RARP)

Routing-Protokolle

Protokolle der Transportschicht (Layer 4)

Transmission Control Protocol (TCP)

User Datagram Protocol (UDP)

Protokolle der Anwendungsschicht (Layer 5–7)

Sonstige Protokolle

Adressierung im IP-Netzwerk

Adresskonzept

Adressierungsverfahren

Adressregistrierung

Adressaufbau und Adressklassen

Subnetzadressierung

Prinzip

Typen und Design der Subnetzmaske

Verwendung privater IP-Adressen

Internetdomain und Subnetz

Dynamische Adressvergabe

Bootstrap Protocol (BootP)

Dynamic Host Configuration Protocol (DHCP)

IP-Version 6 (IPv6)

Gründe für eine Neuentwicklung

Lösungsansätze

IPv6-Leistungsmerkmale

IP-Header der Version 6

Stand der Einführung von IPv6

NAT, CIDR und RSIP als Alternativen

Fazit

Routing

Grundlagen

Aufgaben und Funktion

Anforderungen

Funktionsweise

Router-Architektur

Routing-Verfahren

Routing-Algorithmus

Einsatzkriterien für Router

Routing-Protokolle

Routing Information Protocol (RIP)

RIP-Version 2

Open Shortest Path First (OSPF)

HELLO

Interior Gateway Routing Protocol (IGRP)

Enhanced IGRP

Intermediate System – Intermediate System (IS-IS)

Border Gateway Protocol (BGP)

Betrieb und Wartung

Router-Initialisierung

Out-Of-Band Access

Hardware diagnose

Router-Steuerung

Sicherheitsaspekte

Software Defined Networking (SDN)

Netzwerk Virtualisierung

Switching Fabrics

WAN Traffic Engineering

SD-WAN

Access Networks

Namensauflösung

Prinzip der Namensauflösung

Symbolische Namen

Namenshierarchie

Funktionsweise

Statische Namensauflösung

Dynamische Namensauflösung

Aufgaben und Funktionen

Auflösung von Namen

DNS-Struktur

DNS-Anfragen

Umgekehrte Auflösung

Standard Resource Records

DNS-Message

Dynamic DNS (DDNS)

Zusammenspiel von DNS und Active Directory

Auswahl der Betriebssystemplattform

Namensauflösung in der Praxis

Vorgaben und Funktionsweise

DNS-Konfiguration

Client-Konfiguration

Protokolle und Dienste

TELNET

SSH (Secure Shell)

SSH-Server-Einrichtung

SSH-Client-Einrichtung

Dateiübertragung mit FTP

Funktion

Sicheres FTP (FTPS und SFTP)

Anonymus FTP

Trivial File Transfer Protocol (TFTP)

HTTP

Eigenschaften

Adressierung

HTTP-Message

HTTP-Request

HTTP-Response

Statuscodes

Methoden

MIME-Datentypen

HTTP Version 2 (HTTP/2)

HTTP/3 und QUIC

HTTPS

E-Mail

Simple Mail Transfer Protocol (SMTP)

Post Office Protocol 3 (POP3)

Internet Message Access Protocol 4 (IMAP4)

Unified Collaboration and Communication (UCC)

Presence Manager

Instant Messaging (IM)

Conferencing

Telephony

Application Integration

Mobility

CTI und Call Control

Federation

Lightweight Directory Access Protocol (LDAP)

Konzeption

Application Programming Interface (API)

NFS

Remote Procedure Calls (Layer 5)

External Data Representation (XDR)

Prozeduren und Anweisungen

Network Information Services (NIS) – YELLOW PAGES

Kerberos

Simple Network Management Protocol (SNMP)

SNMP und CMOT – zwei Entwicklungsrichtungen

SNMP-Architektur

SNMP-Komponenten

Structure and Identification of Management Information (SMI)

Management Information Base (MIB)

SNMP-Anweisungen

SNMP-Message-Format

SNMP-Sicherheit

SNMP-Nachfolger

Sicherheit im IP-Netzwerk

Interne Sicherheit

Hardwaresicherheit

UNIX-Zugriffsrechte

Windows- und macOS-Zugriffsrechte

Benutzerauthentifizierung

Die R-Kommandos

Remote Execution (rexec)

Externe Sicherheit

Öffnung isolierter Netzwerke

Das LAN/WAN-Sicherheitsrisiko

Organisatorische Sicherheit

Data Leakage

Nutzung potenziell gefährlicher Applikationen

Prozessnetzwerke und ihr Schutz

Angriffe aus dem Internet

»Hacker« und »Cracker«

Scanning-Methoden

Denial of Service Attack

DNS-Sicherheitsprobleme

Schwachstellen des Betriebssystems

Virtual Private Network (VPN)

Sicherheitsprotokoll IPsec

IPsec-Merkmale

IP- und IPsec-Paketformat

Transport- und Tunnelmodus

IPsec-Protokolle AH und ESP

Internet Key Exchange (IKE)

Weitere Überlegungen

Grundschriftbuch für IT-Sicherheit des BSI

Patching

Der Schutz des Perimeters

Public Key Infrastructure (PKI)

Security Incident und Event Management (SIEM)

Datenschutz-Grundverordnung (DSGVO)

Der Sicherheitsschild

Troubleshooting in IP-Netzwerken

Analysemöglichkeiten

Der Netzwerk-Trace

Netzwerkstatistik

Remote Network Monitoring (RMON)

Analyse in Switched LANs

Verbindungstest mit PING

Selbsttest

Test anderer Endgeräte

Praktische Vorgehensweise im Fehlerfall

Informationen per NETSTAT

ROUTE zur Wegekongfiguration

Wegeermittlung per TRACEROUTE

Knotenadressen per ARP

Aktuelle Konfiguration

NSLOOKUP zur Nameserver-Suche

Netzwerkanalyse mit WireShark

Installation und Konfiguration

Szenario: Web-Surfing

Diverse Auswertungen

Anhang: Das neue TCP/IP-Umfeld

Internet of Things (IoT)

IoT in der Industrie

IoT im öffentlichen Sektor

IoT im privaten Haushalt

Industrie 4.0

Anhang: TCP/IP-Konfigurationen

Microsoft Windows

Apple macOS

Debian Linux

Android

Apple IOS

Index

Netzwerke

Wie wir heute wissen, entwickelte sich die zunächst durch Teilung von Ressourcen begründete Netzwerk-Implementierung in Unternehmen (z.B. durch die Einführung von Abteilungsdruckern oder gemeinsam genutzte Speichermedien) in den letzten zwei Jahrzehnten zunehmend zu einem Umfeld komplexer weltweiter Kommunikation. Dabei wurde bereits recht früh der Grundstein für eine Standardisierung der »Sprache der Kommunikation« gelegt. Mit Gründung des Internets in den frühen 80er Jahren des letzten Jahrhunderts wurde die Netzwerkprotokollfamilie »TCP/IP« ins Leben gerufen. Sie sorgte dafür, dass Daten zwischen Kontinenten, Ländern und einzelnen Standorten nach festgelegten Kriterien übertragen werden konnten. Das Regelwerk besitzt bis heute seine Gültigkeit und basiert im Wesentlichen auf die in diesem Kapitel dargestellten Standards.

Schnell entstanden die ersten lokalen Netzwerke (LAN = *Local Area Network*), die dann nach und nach über entsprechende Anbindungen mehr und mehr zu standortübergreifenden WAN-Netzwerken (WAN = *Wide Area Network*) ausgebaut wurden.

Netzwerkstandards

Wie bei jeder Technologie, so werden auch im Netzwerkbereich bestimmte Vorgaben, Normen oder Standards benötigt, an denen sich die verschiedenen Entwicklungen orientieren. Bei der Behandlung von Netzwerkstandards sind dies insbesondere das ISO/OSI-Referenzmodell sowie die Normierungen und Vorgaben des IEEE *Institute of Electrical and Electronic Engineers* (Institut der elektrischen und elektronischen Ingenieure), aber ebenso auch bestimmte Kommentierungen von Entwicklungen, die unter der Bezeichnung RFC (*Request For Comment*) allgemein üblich sind und vom IETF (*International Engineering Task Force*) verwaltet werden.

OSI als Grundlage

Zur Vereinheitlichung der Datenübertragung wurde das OSI-Referenzmodell geschaffen, das bestimmte Vorgaben für die Kommunikation offener Systeme darlegt. Das OSI-Modell ermöglicht den Herstellern, ihre Produkte für den Netzwerkeinsatz aufeinander abzustimmen und Schnittstellen offenzulegen.

Dies ist somit die Basis für sämtliche Festlegungen im sogenannten ISO- bzw. OSI-Schichtenmodell, wobei die einzelnen Schnittstellen dieser Norm auf insgesamt sieben

Schichten, sogenannte *Layer*, verteilt werden. Jede Schicht wiederum erfüllt bei der Kommunikation eine bestimmte Funktion. Das OSI-Schichtenmodell diente in der Vergangenheit, aber auch heutzutage noch generell als Grundlage für die Kommunikationstechnologie. Dabei liegt der Sinn und Zweck darin, dass die Teilnehmer der Kommunikation (z.B. Rechner) über genormte Schnittstellen miteinander kommunizieren. Im Einzelnen setzt sich das OSI-Referenzmodell aus den folgenden Schichten zusammen:

Schicht 1 – Physical Layer

Auf dem *Physical Layer* (physikalische Schicht) wird die physikalische Einheit der Kommunikationsschnittstelle dargestellt. Diese Schicht (Bit-Übertragungsschicht) definiert somit sämtliche Definitionen und Spezifikationen für das Übertragungsmedium (Strom-, Spannungswerte), das Übertragungsverfahren oder auch Vorgaben für die Pinbelegung, Anschlusswiderstände usw.

Schicht 2 – Data Link Layer

Der sogenannte Data Link Layer (Verbindungsschicht) ermöglicht eine erste Bewertung der eingehenden Daten. Durch Überprüfung auf die korrekte Reihenfolge und die Vollständigkeit der Datenpakete werden beispielsweise

Übertragungsfehler direkt erkannt. Dazu werden die zu sendenden Daten in kleinere Einheiten zerlegt und als Blöcke übertragen. Ist ein Fehler aufgetreten, werden einfach die als fehlerhaft erkannten Blöcke erneut übertragen.

Schicht 3 – Network Layer

Der *Network Layer* (Netzwerkschicht) übernimmt bei einer Übertragung die eigentliche Verwaltung der beteiligten Kommunikationspartner, wobei insbesondere die ankommenden bzw. abgehenden Datenpakete verwaltet werden. In dieser Vermittlungsschicht erfolgt unter anderem eine eindeutige Zuordnung über die Vergabe der Netzwerkadressen, indem der Verbindung weitere Steuer- und Statusinformationen hinzugefügt werden. In einem Netzwerk eingesetzte Router arbeiten immer auf der Schicht 3 des OSI-Referenzmodells.

Schicht 4 – Transport Layer

Auf dem *Transport Layer* (Transportschicht) werden die Verbindungen zwischen den Systemschichten 1 bis 3 und den Anwendungsschichten 5 bis 7 hergestellt. Dies geschieht, indem die Informationen zur Adressierung und zum Ansprechen der Datenendgeräte (z.B. Arbeitsstationen, Terminals) hinzugefügt werden. Aus dem Grund enthält diese Schicht auch die meiste

Logik sämtlicher Schichten. Im Transport Layer wird die benötigte Verbindung aufgebaut und die Datenpakete werden entsprechend der Adressierung weitergeleitet. Somit ist diese Schicht unter anderem auch für Multiplexing und Demultiplexing der Daten verantwortlich.

Schicht 5 – Session Layer

Der *Session Layer* (Sitzungsschicht) ist die Steuerungsschicht der Kommunikation, wo der Verbindungsaufbau festgelegt wird. Tritt bei einer Übertragung ein Fehler auf oder kommt es zu einer Unterbrechung, wird dies von dieser Schicht abgefangen und entsprechend ausgewertet.

Schicht 6 – Presentation Layer

Die Anwendungsschicht (*Presentation Layer*) stellt die Möglichkeiten für die Ein- und Ausgabe der Daten bereit. Auf dieser Ebene werden beispielsweise die Dateneingabe und -ausgabe überwacht, Übertragungskonventionen festgelegt oder auch Bildschirmdarstellungen angepasst.

Schicht 7 – Application Layer

Schicht 7 ist die oberste Schicht des OSI-Referenzmodells (*Application Layer*), auf der die Anwendungen zum Einsatz kommen. Dies ist somit die Schnittstelle zwischen dem System

(z.B. Rechner oder sonstige Hardware) und einem Anwendungsprogramm.

IEEE-Normen

Neben dem ISO-Schichtenmodell existieren weitere Vorgaben oder Normen für den Netzwerkbereich. Eine wichtige Institution ist dabei das IEEE (*Institute of Electrical and Electronics Engineers*).

Das IEEE (gesprochen *Ai Trippel I*) ist ein Berufsverband für Ingenieure und ein amerikanisches Normungsgremium, das sich generell mit Festlegungen, Standards und Normen für die Kommunikation auf den beiden untersten Ebenen des OSI-Schichtenmodells (*Physical Layer, Data Link Layer*) beschäftigt. Die einzelnen Definitionen in Bezug auf die Datenübertragung werden allesamt unter dem Titel des »Komitee 802« zusammengefasst. Eine der ersten Definitionen des Komitees war die Verabschiedung des Ethernet-Zugriffsverfahrens CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*). Im Dezember 1980 trat eine spezielle Projektgruppe (802.5) zusammen, um das Zugriffsverfahren für den Token-Ring-Bereich zu standardisieren. Ein Jahr später konstituierte sich dann die Token-Bus-Projektgruppe (802.4). Die zahlreichen IEEE-Arbeitsgruppen sind verschiedenen Themen gewidmet

und beschäftigen sich vor allem mit Netzwerktopologien, Netzwerkprotokollen oder Netzwerkarchitekturen. Die wichtigsten der Normen und Arbeitsgruppen sind nachfolgend aufgeführt. Details hierzu können aber auch jederzeit auf der Website des IEEE unter www.ieee.org nachgelesen werden.

IEEE 802.1

Der IEEE-Standard mit der Bezeichnung 802.1 beschreibt den Austausch der Daten unterschiedlicher Netzwerke. Dazu gehören Angaben zur Netzwerkarchitektur und zum Einsatz von Bridges (Brücken). Zusätzlich erfolgen hier auch Angaben über das Management auf der ersten Schicht (*Physical Layer*). IEEE 802.1 wird in der Fachliteratur auch mit dem Namen *Higher Level Interface Standard* (HLI) bezeichnet.

IEEE 802.1Q

Innerhalb des Arbeitskreises HLI (*Higher Level Interface*) beschäftigt sich die Arbeitsgruppe 802.1Q mit der Definition des Standards für den Einsatz virtueller LANs (VLANs).

IEEE 802.2

Im Arbeitskreis 802.2 wird eine Definition für das Protokoll festgelegt, mit dem die Daten auf der zweiten Ebene des OSI-Modells (*Data Link*) behandelt werden. Dabei wird

unterschieden zwischen dem verbindungslosen und dem verbindungsorientierten Dienst. Die Einordnung dieses Standards im OSI-Referenzmodell erfolgt auf Schicht 2.

IEEE 802.3

Dies ist eine Definition, die im Bereich der Netzwerke eine der wichtigsten Vorgaben darstellt, denn mit 802.3 wird neben der Topologie, dem Übertragungsmedium und der Übertragungsgeschwindigkeit auch ein ganz spezielles Zugriffsverfahren beschrieben bzw. vorgegeben: CSMA/CD, was als Abkürzung für *Carrier Sense Multiple Access, Collision Detection* steht. Darüber hinaus werden weitere Definitionen festgelegt, die sich allesamt mit dem Einsatz des Übertragungsmediums befassen (10Base-2, 10Base-5, 10Base-T, 100Base-T, Fast Ethernet, Gigabit Ethernet usw.). Der Standard wird in der Ethernet Working Group des IEEE stets erweitert und aktualisiert. Hier einige Beispiele:

IEEE 802.3ab

Diese Arbeitsgruppe spezifiziert die notwendigen Vorgaben, um den Einsatz von Gigabit Ethernet auf UTP-Kabeln (*Twisted Pair*) der Kategorie 5 zu ermöglichen. Die Standardisierung erfolgte im Jahre 1999.

IEEE 802.3ac

Diese Arbeitsgruppe befasst sich mit MAC-Spezifikationen (*Media Access Control*) und Vorgaben für das Management des Ethernet-Basisstandards, inklusive bestimmter Vorgaben für den Einsatz virtueller LANs (VLANs). Die Standardisierung erfolgte im Jahre 1998.

IEEE 802.3an

Im Jahr 2006 wurde der Standard 802.3an verabschiedet, der eine Übertragung von 10 Gbit auf herkömmlichen Kupferkabeln des Typs *Twisted Pair* vorsieht. Beim Einsatz von Cat6-Kabeln können Daten über eine Distanz von 100 Metern übertragen werden, mit Cat5e-Kabeln immerhin noch über eine Distanz von 22 Metern.

IEEE 802.3db

Diese Gruppe innerhalb des neueren Projekts 802.3-2018 beschäftigt sich mit den physischen Spezifikationen und Parametern für den Betrieb von 100-, 200- und 400-Gigabit-Ethernet-Netzwerken via Glasfaser unter Verwendung einer 100-Gigabit-Signalisierung. Dieser Standard wurde am 3. Juni 2020 verabschiedet und hat zunächst eine Laufzeit bis zum 31. Dezember 2024.

IEEE 802.3z

Diese Arbeitsgruppe legt Standards für Gigabit Ethernet fest, insbesondere für den Einsatz von Gigabit Ethernet auf Kupferkabeln der Kategorie 5 (siehe auch 802.3ab), aber ebenso für die Übertragung mittels Glasfaserkabel. Dieser Standard wurde im Jahre 1998 verabschiedet.

IEEE 802.4

Während sich 802.3 mit Ethernet beschäftigt, wird in der Definition 802.4 der Token-Bus-Standard proklamiert und entsprechende Festlegungen werden getroffen.

IEEE 802.5

Als Ergänzung zu 802.4 legt dieser Arbeitskreis eine Definition für den Token Ring fest. Dazu zählt die Definition der Topologie, des Source Routing, des Übertragungsmediums und auch der Übertragungsgeschwindigkeit.

IEEE 802.6

Dieser Standard beschreibt den Einsatz von MANs, also die sogenannten *Metropolitan Area Networks*. Zusätzlich beschäftigt sich diese Gruppe auch mit dem Bereich der DQDB-Protokolle (*Distributed Queue Dual Bus*).

IEEE 802.7

Mit den Festlegungen innerhalb dieser Arbeitsgruppe (*Broadband Technical*) werden bzw. wurden Vorgaben für den Einsatz der Breitbandtechnologie festgelegt.

IEEE 802.8

802.8 beschäftigt sich ausschließlich mit dem Einsatz von Lichtwellenleitern bzw. Glasfaserkabeln (*Fiber Optic*) innerhalb eines Netzwerks.

IEEE 802.9

Die Inhalte dieses Standards beziehen sich auf die Einbeziehung von Sprachübertragung in die allgemeine Kommunikation. Auf diese Art und Weise sollen in einem solchen ISLAN (*Integrated Services LAN*) alle Datenendgeräte (Rechner, Drucker, Telefon, Fax usw.) an einer einzigen Schnittstelle betrieben werden können.

IEEE 802.9a

Die isochrone Technik für die Echtzeitübertragung von Daten im LAN bis an den Arbeitsplatz ist Inhalt dieser Arbeitsgruppe.

IEEE 802.10

Der Arbeitskreis mit der Bezeichnung 802.10 beschäftigt sich vornehmlich mit generellen Sicherheitsfragen. Zu diesem Zweck wurde auch eine entsprechende Vorgabe verabschiedet, die den Namen SILS trägt (*Standard for Interoperable LAN Security*).

IEEE 802.11

Die in 802.11 zusammengesetzte Projektgruppe beschäftigt sich mit dem Einsatz drahtloser LANs (WLAN). Auf die einzelnen Basis-Standards dieser Projektgruppe wird in [Abschnitt 1.2.2](#) näher eingegangen.

IEEE 802.12

Ergebnis dieser Arbeitsgruppe ist ein Standard für ein 100-Mbit-Verfahren für den Multimedia-Einsatz, das den Namen *Demand Priority* (DP) trägt. Es handelt sich dabei um ein Zugriffsverfahren (vergleichbar mit CSMA/CD aus 802.3), bei dem ein Repeater die einzelnen Datenendgeräte nach Übertragungswünschen abfragt (Polling-Verfahren).

IEEE 802.14

Der Auftrag der 802.14-Arbeitsgruppe besteht bzw. bestand darin, Standards und Normen für den Bereich von Kommunikationsfunktionen in Kabelnetzen (Kabelfernsehen) auszuarbeiten. Diese Bestrebungen sind in der Literatur auch häufig unter der Abkürzung CATV (*Cable Television*) zu finden.

IEEE 802.15

Die kabellose Anbindung von Rechnern ist Auftrag dieser Arbeitsgruppe. In Erweiterung zur Arbeitsgruppe 802.11, die sich mit drahtlosen LANs (WLANs) beschäftigt, wird in dieser Gruppe die Gesamtheit der kabellosen Anbindungsmöglichkeiten auf Basis des WPAN (*Wireless Personal Area Network*) betrachtet. Darunter fallen beispielsweise Technologien für den kabellosen Einsatz auf kurzen Distanzen (z.B. Bluetooth, ZigBee).

IEEE 802.16

Als Ergänzung zur Arbeitsgruppe 802.15 beschäftigt sich diese Arbeitsgruppe mit der kabellosen Anbindung in der Breitbandtechnik. Bekannt geworden ist diese Technik unter dem Namen WIMAX (*Worldwide Interoperability for Microwave*

Access), die als Alternative zum Festnetz-DSL angesehen werden kann.

Netzwerkvarianten

Neben Festlegungen, Standards und Normen entscheiden letztlich der Anwender und natürlich die Industrie über entsprechende Produktpaletten, also welche Formen der Netzwerkvarianten zum Einsatz kommen. Heutzutage kann man festhalten, dass bei sämtlichen Neuinstallationen fast durchweg der Netzwerktyp *Ethernet* verwendet wird. Was es damit auf sich hat und welche sonstigen Varianten darüber hinaus zur Verfügung stehen (Token Ring, ATM usw.), wird in [Abschnitt 1.2.4](#) in einer Übersicht dargestellt.

Während sich Ethernet und Token Ring als etablierte LAN-Standards im Laufe der letzten Jahrzehnte in den Unternehmen als verlässliche Kommunikationsgebilde durchgesetzt haben und FDDI bzw. ATM als Hochgeschwindigkeitstechnologie mit hohen Übertragungskapazitäten in Backbones eingesetzt wurden, wurden Fast Ethernet und Gigabit Ethernet für Hochgeschwindigkeits-LANs mit gewachsenen Anforderungen immer bedeutsamer und sind in zahlreichen Netzwerken bereits vollständig implementiert.

Noch Ende des vergangenen Jahrhunderts reichten Ethernet-Kapazitäten von 2 bis 10 Mbit/s und Token-Ring-Geschwindigkeiten von 4 bis 16 Mbit/s für den anfallenden Datenverkehr völlig aus. Diese Situation hat sich mittlerweile jedoch deutlich verändert, da die Übertragung multimedialer Objekte wie Bilder, Grafiken, Video- und Audiosequenzen in Netzwerken immer wichtiger geworden ist. Netzwerkstrukturen, die unter den Schlagworten *Corporate Networking*, *Voice over IP* oder *Videoconferencing* zusammengefasst werden, tragen dazu bei, höchste Bandbreiten im lokalen Netzwerk zur Verfügung stellen zu müssen, sodass der Bedarf an Hochgeschwindigkeitstechnologien wie dem Gigabit Ethernet mit Kapazitäten von 1.000, 10.000 Mbit/s und mehr nur eine Frage der Zeit war.

Zwar entwickelte man auch für die Token-Ring-Technik Komponenten mit höherer Leistung (*High Speed Token Ring* = HSTR), es zeigte sich aber, dass der Markt diese Technik nur dann annahm, wenn ein Unternehmen, das bereits primär Token-Ring-Netzwerke einsetzte, auf eine deutlich höhere LAN-Geschwindigkeit umstellen musste und den Wechsel zu Ethernet nicht vornehmen wollte. Heute ist die Token-Ring-Technologie auf globaler Basis weitgehend verschwunden und wird auch nicht mehr weiterentwickelt, sodass in diesem Buch

fast ausschließlich Ethernet als Netzwerkvariante im Detail erläutert wird.

Ethernet

Durch Einsatz eines speziellen Zugriffsverfahrens mit dem Namen CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) verdichtete sich bereits in den 70er Jahren des vorigen Jahrhunderts der Ethernet-Standard. Dabei repräsentiert Ethernet einen Standard, der physikalisch auf einer reinen Bus-Topologie beruht. Diesen Bus kann man sich als ein Kabel vorstellen, das an seinen beiden Enden durch jeweils einen Abschlusswiderstand terminiert wird (Terminator) und über sogenannte Transceiver dem jeweiligen Endgerät (z.B. Rechner mit Ethernet-Netzwerkcontroller) einen Netzwerkzugang ermöglicht.

Auch wenn es für die Hochgeschwindigkeitstechnologien alternative LAN-Zugriffsverfahren bzw. entsprechende Entwicklungen gab (z.B. 100VG-AnyLAN), hat sich heutzutage Ethernet mit dem CSMA/CD-Verfahren als Grundlage und Standard durchgesetzt. Dabei beruht das CSMA/CD-Verfahren auf folgenden Überlegungen:

Eine Station (Rechner in einem lokalen Netzwerk) möchte Daten übertragen. Zu diesem Zweck versucht sie, über die

eingebaute Netzwerkkarte auf dem Übertragungsmedium zu erkennen, ob eine andere Station Daten überträgt (*Carrier Sense*). Wenn das Medium besetzt ist (*Collision Detection*), zieht sich die Station wieder zurück und wiederholt diesen Vorgang in unregelmäßigen Abständen, bis die Leitung frei ist. Dann beginnt sie mit dem Übertragungsvorgang. Alle am Netz befindlichen Rechner überprüfen den *Header* des ankommenden Datenpakets (*Frame*), und nur derjenige, dessen eigene Adresse mit der Zieladresse im Frame übereinstimmt, beginnt mit dem Empfangsprozess (siehe [Abb. 1-1](#)).

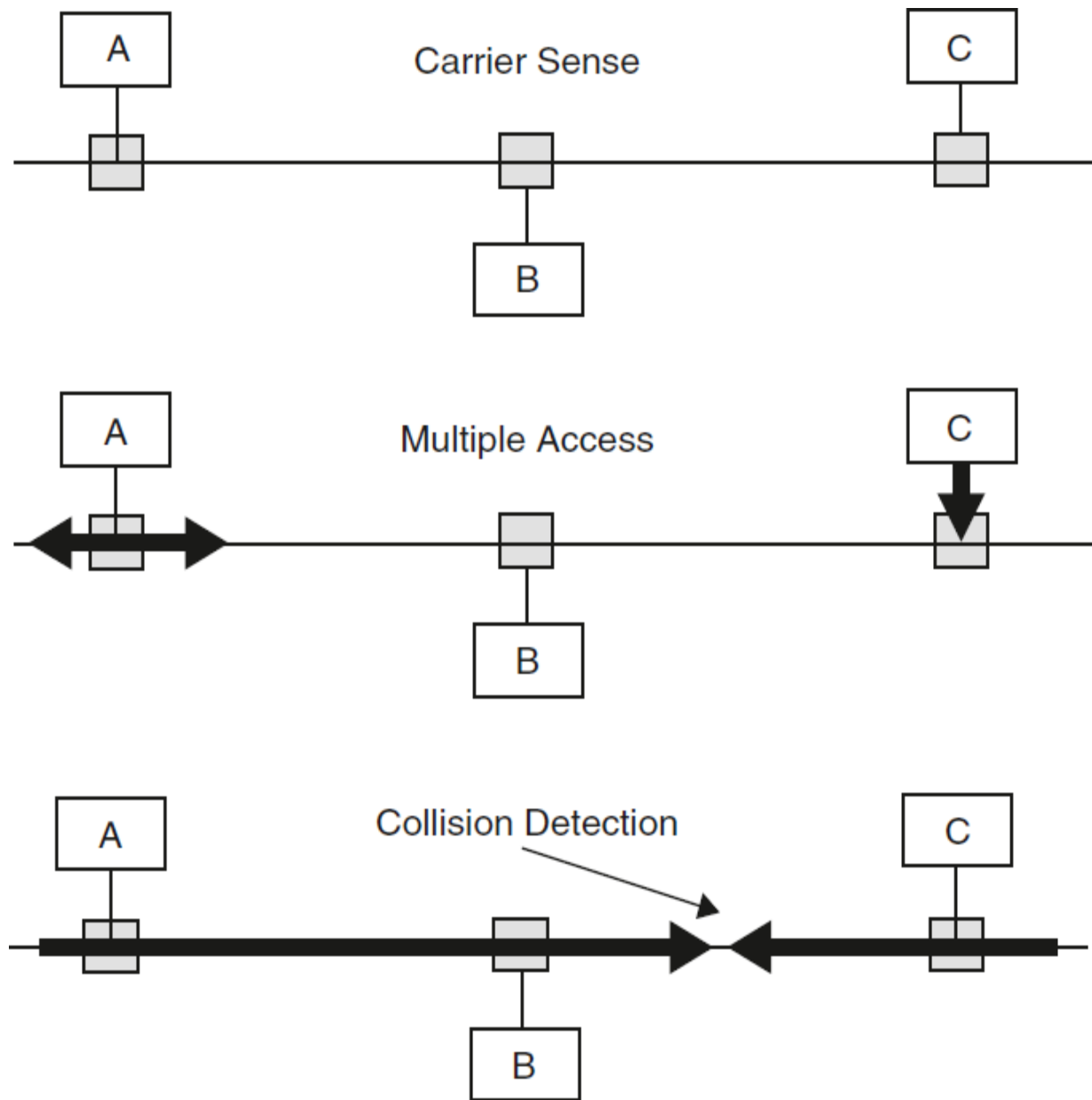


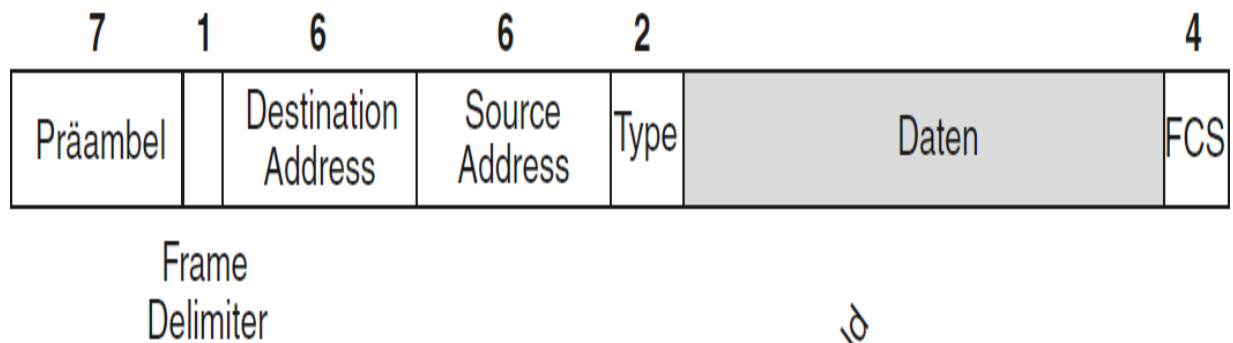
Abb. 1–1 CSMA/CD-Zugriffsverfahren im Ethernet

Dabei ist bei der gleichzeitigen Übertragung mehrerer Stationen zu beachten, dass das Prinzip der Kollisionserkennung (*Collision Detection*) dazu führt, dass die erste sendende Station ihren Sendeprozess unmittelbar abbricht und ein Störsignal (Jam-Signal) produziert. Ein

erneuter Sendeversuch wird innerhalb zufällig generierter Intervalle wiederholt. Zufällig gebildete Intervalle minimieren das Risiko überproportional steigender Kollisionen. Kommt nach weiteren Sendeversuchen mit unterschiedlich langen Wartezeiten und fünf weiteren, gleich großen Zeitintervallen keine störungsfreie Übertragung zustande, wird die nächsthöhere Protokollschicht informiert und muss nun ihrerseits geeignete Sicherungsmechanismen durchführen.

Bei der Betrachtung des Ethernet-Standards (Version 2) ist zu berücksichtigen, dass dieser leicht vom 802.3-Standard abweicht. Die Differenzen zeigen sich insbesondere in unterschiedlichen maximalen Signalrundlaufzeiten und im Aufbau der Datenpakete (*Frames*). So befindet sich im Ethernet-Frame auf Byte-Position 20 ein Zwei-Byte-Typenfeld, aus dem das hier eingesetzte höhere Protokoll hervorgeht. Anschließend beginnt der Datenteil. Im IEEE-802.3-Frame hingegen fehlt das Typenfeld. Stattdessen gibt es ein gleich großes Längenfeld, in dem die Gesamtlänge des Frames eingesetzt wird. Anschließend folgt der LLC-Header (*Logical Link Control*) mit den Daten. Daraus ergibt sich eine Inkompatibilität von Rechnern, die mit diesen beiden Standards arbeiten und miteinander kommunizieren wollen. In [Abbildung 1–2](#) sind die Unterschiede im Frame-Aufbau dargestellt.

a) Ethernet Frame Format



b) IEEE 802.3 Frame Format

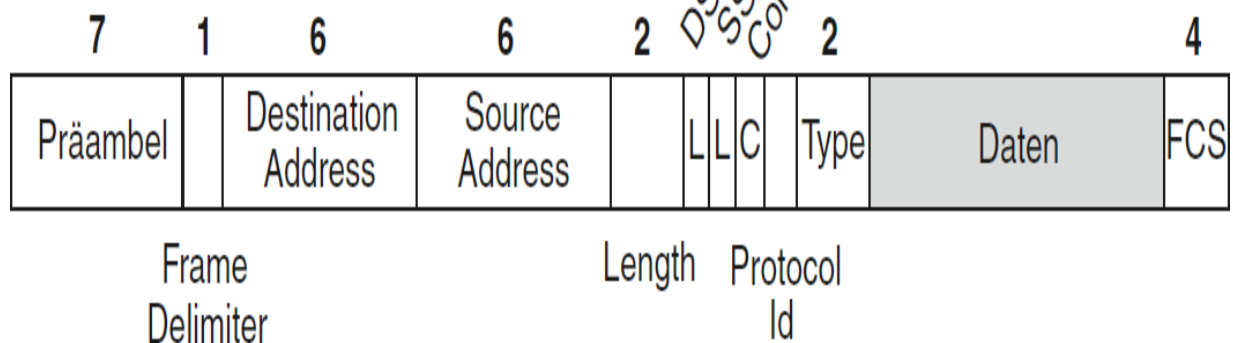


Abb. 1–2 Unterschiede im Aufbau der Datenpakete

Fast Ethernet

Fast Ethernet beschreibt einen als IEEE 802.3u definierten Standard, der aus dem klassischen Ethernet hervorgegangen ist. Seine Implementierung wird als 100BaseT bezeichnet und stellt eine Bandbreite von 100 Mbit/s zur Verfügung. 100BaseT beruht auf dem IEEE-802.3-Standard und ist somit in der Lage, beide Geschwindigkeiten, also sowohl 10 Mbit/s als auch 100 Mbit/s, im lokalen Netzwerk zu realisieren. Auch das Frame-Format ist für beide Implementierungen identisch.

Da 100BaseT das gleiche Zugriffsverfahren wie 10BaseT verwendet (CSMA/CD), ist eine Reduktion der als *Collision Domain* bezeichneten Entfernung zwischen zwei Ethernet-Stationen von etwa 2000 Metern auf 200 Metern erforderlich. Das Design einzelner Netzwerksegmente ist abhängig von den für dieses Verfahren eingesetzten Medien. 100BaseTX-, 100BaseFX- und 100BaseT4-Medien unterscheiden sich jeweils im Kabeltyp, in der Anzahl einzelner Adern und in den verwendeten Anschlusstypen. Fast Ethernet hat in den letzten Jahren jedoch deutlich an Bedeutung verloren und wird zunehmend durch die neueren Gigabit-Standards ersetzt. Dies gilt nicht nur für industrielle Netzwerke, sondern auch für die eingesetzten Netzkomponenten im privaten Bereich.

Gigabit Ethernet

Die unmittelbare Weiterentwicklung von Fast Ethernet stellt Gigabit Ethernet dar. Aus Sicht des ISO/OSI-Layers 2 (*Data Link Control*) in Richtung höherer Protokollschichten ist die Architektur von Gigabit Ethernet mit dem IEEE-802.3-Standard identisch. Allerdings mussten die 1999 neu geschaffenen Standards IEEE 802.3z (Gigabit Ethernet über Glasfaser) und IEEE 802.3ab (Gigabit Ethernet über Kupferkabel 1000BaseT) hinsichtlich der physikalischen Schicht angepasst werden,

damit Geschwindigkeiten von bis zu 1000 Mbit/s erreicht werden können.

Die leicht modifizierte Gigabit-Ethernet-Architektur geht aus [Abbildung 1–3](#) hervor. Eine medienunabhängige Schnittstelle (GMII = *Gigabit Media Independent Interface*) innerhalb der physikalischen Schicht sorgt für Transparenz gegenüber den höheren Protokollschichten.

Das Medium spielt beim Gigabit Ethernet eine herausragende Rolle. Glasfaserkabel (Medien 1000BaseCX, 1000BaseLX und 1000BaseSX) sind aufgrund ihrer Materialbeschaffenheit und der anwendbaren Codierungsverfahren für solch hohe Geschwindigkeiten besonders gut geeignet. Aber auch die am 1000BaseT orientierten Twisted-Pair-Kabel lassen sich für das Gigabit Ethernet verwenden, vorausgesetzt, es werden alle vier Adernpaare für die Signalübermittlung verwendet (für 10BaseT oder 100BaseT sind zwei der vier Adernpaare ausreichend). Während der Einsatz des Glasfaserkabels Übertragungsstrecken bis zu 5000 Metern erlaubt, verringert sich die Distanz beim 1000BaseT, also beim Kupferkabel, auf etwa 100 Meter. Dieses ist daher lediglich zur Überwindung kurzer Entfernungen bzw. innerhalb von Verteilerschränken verwendbar.

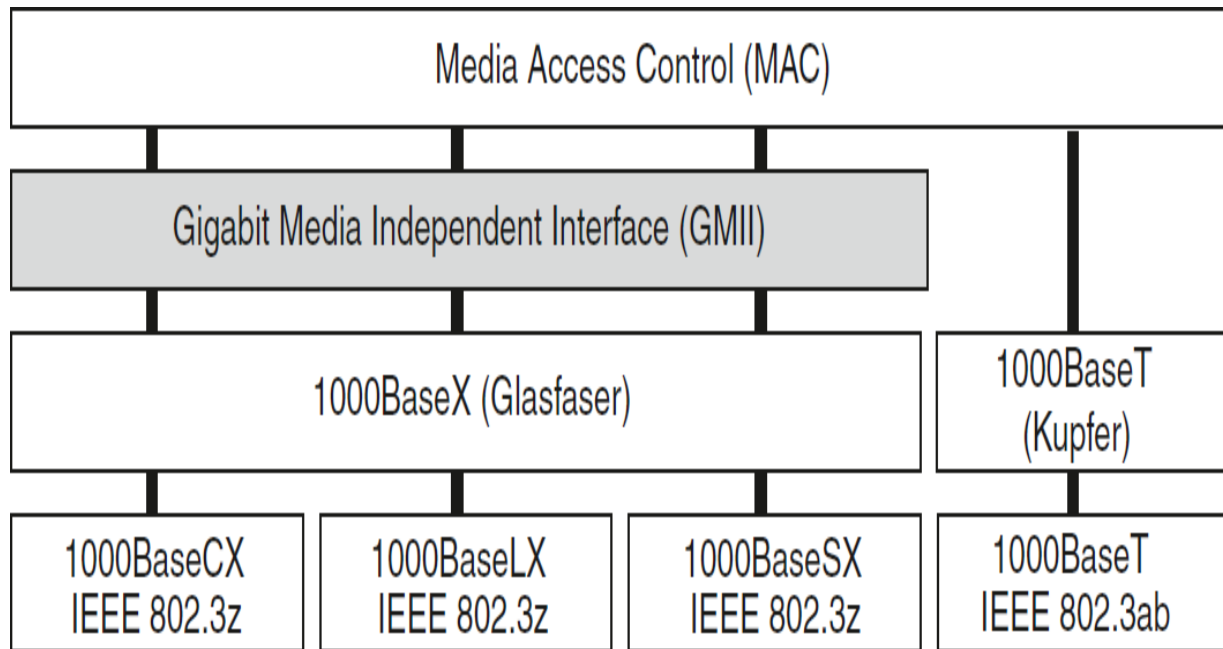


Abb. 1–3 *Darstellung der modifizierten Architektur bei Gigabit Ethernet*

Heute beschäftigen sich technische Arbeitsgruppen (z.B. im IEEE im Projekt 802.3 df) bereits mit der Spezifikation und Implementierung von 800 Gigabit Ethernet (GbE) oder gar 1,6 Terabit Ethernet (TbE).

100VG-AnyLAN

Parallel zur Entwicklung von Fast Ethernet wurde im letzten Jahrhundert die 100VG-AnyLAN-Technologie geschaffen, die alternativ andere LAN-Zugriffsverfahren als das CSMA/CD nutzen kann.

Im IEEE-802.12-Standard ist das von Hewlett Packard entwickelte 100VG-AnyLAN als Alternative zum CSMA/CD-Zugriffsverfahren definiert und kann gewissermaßen als Synthese aus Ethernet und Token Ring betrachtet werden; weder Token Ring noch 100VG-AnyLAN haben heutzutage jedoch noch eine praktische Bedeutung. Der LAN-Zugriff wird im Wesentlichen vom angeschlossenen Hub oder Switch bestimmt und vermeidet zufällig orientierte Zugriffszyklen wie beim CSMA/CD-Verfahren. Konkurrierende Übertragungsanfragen werden vom Hub nach Prioritäten abgewickelt (nur eine einzige Station kann zu einem bestimmten Zeitpunkt Daten übertragen – Kollisionen werden somit vermieden), sodass sich diese Technologie besonders für Multimedia-Datenverkehr eignet. Es besteht eine Frame-Kompatibilität zu Ethernet- und Token-Ring-Netzwerken, sodass ein 100VG-Any-LAN-Segment einfach durch Installation einer Bridge in bestehende Ethernet- oder Token-Ring-Segmente integriert werden kann.

HINWEIS

Die Netzwerkvariante 100VG-AnyLAN hat heutzutage so gut wie keine Bedeutung mehr und kommt insbesondere bei Neuinstallation nicht mehr zum Einsatz.

Wireless LAN (IEEE 802.11)

Überall dort, wo eine Verkabelung zur Einrichtung eines Netzwerks nicht infrage kommt oder besondere Anforderungen hinsichtlich der Mobilität vorliegen, ist der Einsatz funkbasierter Kommunikationstechnologien zu empfehlen. Die heute verfügbare Technik liefert ausreichend Möglichkeiten, um die in einem Netzwerk erforderlichen Aufgaben zufriedenstellend zu lösen. Nachfolgend sind die verfügbaren Standards, Komponenten und die erforderlichen Sicherheitsmechanismen erläutert.

WLAN-Standards

Im Standard IEEE 802.11, der in seinen Einzelstandards die heute verfügbaren Techniken darstellt, sind die wesentlichen Merkmale und Funktionen erläutert und definiert. Hier einige wichtige Beispiele:

- *IEEE 802.11a*

WLAN-Standard (1999), der auf einer Brutto-Datenrate von etwa 54 Mbit/s und dem 5-GHz-Band basiert. Seit November 2002 ist dieser Standard auch für Deutschland zugelassen (gem. Order Nr. 35/2002 vom 13. November 2002 – Regulierungsbehörde für Telekommunikation und Post).

- *IEEE 802.11b*

WLAN-Standard (1999), der auf einer Brutto-Datenrate von etwa 11 Mbit/s und dem 2,4-GHz-Band basiert. Es handelt sich hierbei um die derzeit verbreitetste WLAN-Variante.

- *IEEE 802.11g*

WLAN-Standard aus dem Jahr 2003, der auf einer Brutto-Datenrate von etwa 54 Mbit/s und dem 2,4-GHz-Band basiert. Dieser Standard ist abwärtskompatibel zu IEEE 802.11b, sodass hier ein zukünftig hoher Verbreitungsgrad sehr wahrscheinlich ist.

- *IEEE 802.11h*

WLAN-Standard, der eine Adaption des IEEE-802.11a-Standards für Europa repräsentiert. In Europa erfolgt die Radar-Kommunikation ebenfalls im 5-GHz-Band, sodass hier besondere Vorkehrungen zur Steuerung der Datenübertragung vorgenommen werden müssen.

- *IEEE 802.11n (Wi-Fi 4)*

Der sogenannte »n-Standard« 802.11n wurde Ende 2009 verabschiedet und ermöglicht eine Brutto-Datenrate von theoretisch bis zu 600 Mbit/s. Die Übertragung erfolgt im 2,4-GHz-Band und optional auch unter 5 GHz.

- *IEEE 802.11ac (Wi-Fi 5)*

IEEE 802.11ac stellt die Weiterentwicklung des 802.11n-Standards dar und sorgt für eine Verbesserung der

theoretischen Übertragungsgeschwindigkeit auf mehr als das Doppelte (1,2 Gbit/s). Außerdem liefert es eine weitaus stabilere Datenübertragung auf hohem Niveau. Diese Stabilität konnte der Vorgänger Wi-Fi 4 nicht bieten, da er nicht über das Beam-Forming-Verfahren verfügt und somit eine gezielte Ausrichtung des Funksignals auf Endgeräte und Access Points unter Verwendung mehrerer (mindestens zwei) Antennen möglich ist.

- *IEEE 802.11ax (Wi-Fi 6)*

Der seit Anfang 2020 etablierte Standard IEEE 802.11ax liefert im 2,4-GHz- und 5-GHz-Frequenzband mittlerweile eine vielfach höhere Datenübertragungsrate von theoretisch 10 Gbit/s und mehr, da der Datenstrom parallel mehrere Geräte erreichen kann. Die theoretischen Geschwindigkeiten sind allerdings nur unter Laborbedingungen realisierbar. In der Praxis sorgen sich gegenseitig störende Router und nicht immer optimale Signalqualitäten für deutliche Performance-Einbrüche; insbesondere für vom Router entfernte Endgeräte (Wi-Fi 6 benötigt zur Ausnutzung seiner erhöhten Geschwindigkeiten ein sehr gutes Signal).

- *IEEE 802.11ad*

IEEE 802.11ad nutzt ein völlig anderes Frequenzband als die zuvor dargestellten Standards: das 60-GHz-Band. Hier sind sehr hohe Geschwindigkeiten von bis zu 7 Gbit/s möglich und

durch die Frequenzbandunabhängigkeit ist eine Kommunikation gegenüber Störungen kaum anfällig. Allerdings ist die Reichweite der Kommunikation auf etwa 9 Meter beschränkt, sodass dieser Standard eher innerhalb von Räumen bzw. eng umfassten Bereichen eingesetzt wird (z.B. bei der Übertragung von hoch aufgelösten Videos, wie 4K). Dieser Standard wird gemeinsam mit dem IEEE 802.11ay auch als *Wireless Gigabit* bezeichnet.

- *IEEE 802.11ah*

Die Besonderheit dieses Funkstandards ist seine erhöhte Reichweite. Er wird im 900-MHz-Band betrieben und liefert Datenkommunikation bis zu 1 km Entfernung allerdings mit einer niedrigen Datenrate von mindestens 100 kbit/s. Dieser Standard wird auch als Low-Power-Wi-Fi oder offiziell auch Wi-Fi HaLow bezeichnet. Er kommt primär bei der Datenkommunikation im SmartHome-Bereich (auch in IoT – *Internet of Things* – Umgebungen) zum Einsatz, da der geringe Energieverbrauch und die Robustheit gegenüber Hindernissen und »dicken Betonwänden« einen gegenüber anderen WLAN-Standards deutlichen Vorteil bedeuten. Dies ist allerdings derzeit nur Theorie. Stand Anfang 2022 gibt es kaum Produkte für diesen Standard auf dem Markt, auch wenn von der Wi-Fi-Alliance für die nächsten Jahre vielversprechende Erfolge prognostiziert werden (Wi-Fi-

Alliance-Veröffentlichung vom 2. November 2021, Austin, Texas: *WiFi Certified HaLow delivers long range, low power WiFi*).

- *IEEE 802.11ay*

Als Erweiterung und Verbesserung des IEEE-802.11ad-Standards wurde IEEE 802.11ay im März 2021 mit einer deutlich höheren Datenrate von 20–40 Gbit/s verabschiedet. Reichweite: ca. 300 bis 500 Meter.

- *IEEE 802.11be (Wi-Fi 7)*

Dieser Standard stellt die Weiterentwicklung des Wi-Fi 6 dar und soll u.a. eine bis zu vier Mal höhere Datenrate ermöglichen. Man erwartet somit Datenübertragungsraten von 30 bis 40 Gbit/s. Ein weiterer Fokus liegt auf datenintensiven Echtzeit-Anwendungen zum Beispiel im Bereich Virtual oder Extended Reality. Damit wird Wi-Fi 7 zu einer ersten ernst zu nehmenden Alternative zum Festnetz-Ethernet. Derzeit erwartet man allerdings eine Standardisierung erst für das Jahr 2024.

HINWEIS

Sämtlichen heute im Einsatz befindlichen WLAN-Standards ist gleich, dass es sich um ein *Shared Medium* handelt; dies bedeutet, dass sich die angegebene Bandbreite jeweils nur auf einen einzigen Netzknoten bezieht. Nutzen mehr als ein

Knoten (z.B. mehrere Rechner) das Netzwerk, so muss die verfügbare Bandbreite geteilt werden. Zusätzlich reduzieren Störeinflüsse im Funknetz die Übertragungsqualität der Daten, sodass in einem WLAN in der Praxis beispielsweise bei mehreren miteinander kommunizierenden Rechnern (je nach genutztem Standard) von deutlich reduzierten Übertragungsraten ausgegangen werden muss.

Komponenten

In einem WLAN spielt der Begriff »Zelle« eine entscheidende Rolle. Eine Zelle bezeichnet die Zusammenfassung einzelner im WLAN miteinander kommunizierender Netzwerkknoten. Die Identifikation einer Zelle erfolgt über den sog. ESSID (*Extended Service Set Identifier*). Jeder Zelle sollte im WLAN ein eigener Funkkanal zugeordnet werden.

In einem WLAN lassen sich grundsätzlich zwei verschiedene Topologieformen antreffen:

- *Ad-hoc-Modus*

Dem Ad-hoc-Modus liegt eine direkte Kommunikation der einzelnen Netzwerkknoten zugrunde. Grundsätzlich kommuniziert hier jeder mit jedem und verwendet dabei die gemeinsam zugeordnete ESSID. Eine solche Topologie ist zumeist dann vorteilhaft, wenn lediglich kleine

überschaubare Netzwerke (z.B. für temporäre Arbeitsgruppen oder Besprechungen) gebildet werden sollen. Eine Ad-hoc-Topologie ist mit dem übrigen Netzwerk nicht verbunden und stellt somit eine autonome Kommunikationseinheit dar. In einer Sonderform des Ad-hoc-Netzwerks ermöglicht ein nach IEEE 802.11s konzipiertes WLAN Mesh (vermaschtes Funknetz) die »Selbst-Organisation« von Netzwerk-Teilnehmern (zumeist mobile Endgeräte wie Handys, Tablets, Sensoren, Smart Devices usw.).

- *Infrastruktur-Modus*

Der Topologie-Infrastruktur-Modus stellt in einem WLAN den Regelfall dar. Hier werden sogenannte *Access Points* benötigt, die den Zugang zum kabelbasierten Netzwerk herstellen und somit die Funkzelle in das übrige Unternehmensnetzwerk integrieren. Der Infrastruktur-Modus ermöglicht die Nutzung gemeinsamer Netzwerkressourcen (Festplatten, Drucker, Kommunikationseinrichtungen usw.), siehe hierzu auch

[Abbildung 1–4.](#)

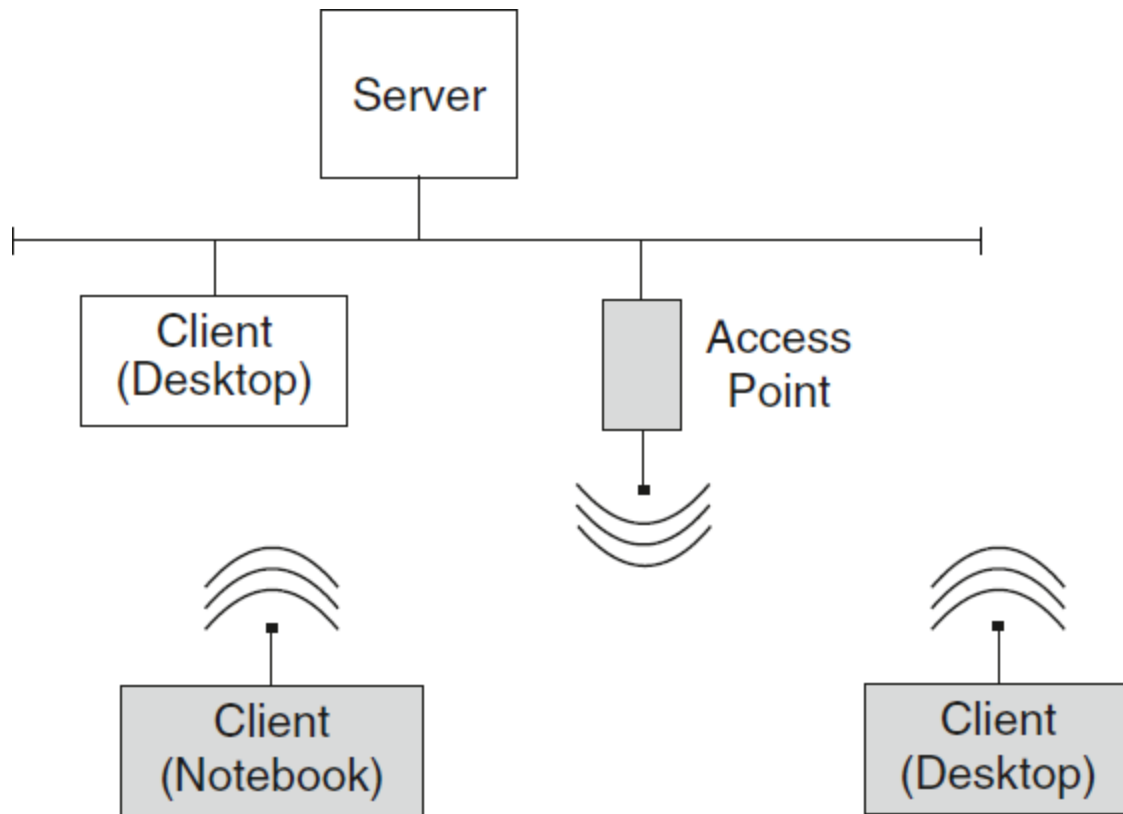


Abb. 1–4 WLAN-Betrieb im Infrastruktur-Modus

Neben den unterschiedlichen Topologien gibt es in einem WLAN weitere charakteristische Komponenten oder Merkmale, die nachfolgend erläutert werden.

- *Access Point*

Wie bereits erwähnt, kommen Access Points lediglich in Infrastruktur-Modus-Netzwerken zum Einsatz; sie stellen dort die Schnittstelle zwischen Funknetz und dem kabelgebundenen Netzwerk dar. Der gesamte Datenverkehr der Funkzelle wird über einen Access Point geführt, sofern mit dem Kabelnetz kommuniziert werden soll. Er ist

Medienübergang, Brücke und Schnittstelle zwischen den Netzwerken. Zahlreiche Router- oder Switch-Hersteller bieten ihre Geräte bereits in einer Access-Point-Variante an.

- *WLAN-Controller*

Jeder Netzknoten muss zur Funkkommunikation in einem WLAN mit einem WLAN-Controller ausgestattet sein. Je nach Endgerät handelt es sich um eine PC-Steckkarte, einen USB-Dongle oder um eingebaute Komponenten, die fest mit der übrigen Hardware verbunden sind. Mittlerweile werden heute Notebooks, Tablets oder auch Mobilfunkgeräte mit eingebauten WLAN-Controllern ausgestattet, sodass der separate Einsatz von Zusatzkarten nicht mehr erforderlich ist.

- *Antennen*

WLAN-Controller und Access Points besitzen jeweils unterschiedlich ausgeprägte Antennen, die für den Versand und Empfang der Daten zuständig sind. Je nach Hardwareerfordernissen handelt es sich dabei um Flachantennen, kleine Stabantennen oder größere Richtantennen, um die Funkreichweite zu vergrößern.

WLAN-Charakteristik

Die Schicht *Physical Layer* im ISO/OSI-Referenzmodell auf Schicht 1 wird beim IEEE-802.11-Standard durch einen PMD-

und einen PLCP-Sub-Layer dargestellt. Während der PMD (*Physical Medium Dependent Sub-Layer*) für die Signalmodulation und -codierung zuständig ist, bildet das PLCP (*Physical Layer Convergence Protocol*) die allgemeingültige Schnittstelle zur übergelagerten Steuerungsschicht (*MAC Layer*).

Bei der physikalischen Datenübertragung werden in Funknetzen nach IEEE 802.11 zur Erhöhung der Datensicherheit sogenannte *Spread-Spectrum-Technologien* eingesetzt. Die beiden Verfahren lauten FHSS (*Frequency Hopping Spread Spectrum*) und DSSS (*Direct Sequence Spread Spectrum*).

Bei FHSS wechseln Sender und Empfänger zyklisch die benutzte Frequenz, wobei sie die gleiche Reihenfolge einhalten (*Hopping Sequence*). Die durch andere Sender verursachten Störungen werden minimiert, da nur diejenigen Sender und Empfänger miteinander kommunizieren können, die auch mit gleicher Hopping Sequence arbeiten. Bei diesem Verfahren ist allerdings der technische Aufwand auf der MAC-Ebene recht hoch, da die Frequenzwechsel gesteuert und synchronisiert werden müssen. Aufgrund besserer Leistungswerte hat sich jedoch DSSS – insbesondere für IEEE 802.11b – durchgesetzt. Im Gegensatz zu FHSS wird bei DSSS das Signal nicht zeitlich

versetzt auf verschiedenen Kanälen versendet, sondern als schmalbandiges Signal durch Multiplexe mit einem PN-Code (*Pseudo-Noise*) direkt in ein breitbandiges Signal umgewandelt. Da dadurch die Sendeintensität unter die Rauschgrenze abgesenkt wird, kann das Signal nur noch von Stationen empfangen werden, die mit dem gleichen PN-Code arbeiten.

Der MAC-Layer weist bei 802.11 eine enge Verwandtschaft mit der kabelgebundenen Variante 802.3 auf. Allerdings muss der drahtlose Standard auf die Besonderheiten der Übertragungsstrecke Rücksicht nehmen. Aufgrund der Signalcharakteristik entfällt die Möglichkeit zum Erkennen von Kollisionen. Daher greift 802.11 auf eine Zugangskontrolle nach dem CSMA/CA-Verfahren zurück.

CSMA/CA steht für *Carrier Sense Multiple Access with Collision Avoidance*. Es beschreibt den Zugriff auf einen gemeinsamen Kanal, indem dieser auf das Vorhandensein eines Trägersignals getestet wird. Der Unterschied zu IEEE 802.3 besteht in der *Collision Avoidance* (Kollisionsvermeidung), deren Implementierung als DCF (*Distributed Coordination Function*) oder PCF (*Point Coordination Function*) realisiert wird. Das DCF-Verfahren wird insbesondere im *Ad-hoc-Modus* eingesetzt, während das PCF-Verfahren im Infrastrukturbetrieb (Verwendung von Access Points, die beispielsweise zur

Koordination der Kanalvergabe verwendet werden) zur Anwendung kommt.

Für eine ungefähre Einschätzung der möglichen Reichweiten (z.B. bei IEEE 802.11b) ist es wenig sinnvoll, theoretische Werte zugrunde zu legen. Danach wäre bei einer Netto-Datenrate von etwa 2 Mbit/s eine Reichweite von etwa 400 Metern im Freien und etwa 45 Metern in geschlossenen Räumen realisierbar. Tatsächlich kann davon ausgegangen werden, dass in einer »normalen Bürolandschaft« bei der Durchquerung einer Etagendecke und dreier weiterer Wände (Entfernung ca. 5 Meter Luftlinie) ein Netto-Datendurchsatz von vielleicht 500 bis 700 kbit/s erreicht werden kann. Ähnliche Werte erhält man auch bei einer Messung auf derselben Etage bei Durchquerung von drei Raumwänden. Es ist daher festzustellen, dass die theoretischen Werte in Realumgebungen stark relativiert werden müssen. Eine wesentlich günstigere Umgebung (z.B. bei freier Fläche) ist dagegen eher selten anzutreffen.

Im Unterschied zum kabelbasierten Netzwerk, in dem störende Einwirkungen von außen heute relativ selten auftreten, ist das Funknetz bei der Datenübertragung zahlreichen Einflüssen ausgesetzt. So ist beispielsweise die Verwendung von Metall in Gebäuden für ein WLAN äußerst

hinderlich, da die Dämpfung der Funkwellen eine Kommunikation deutlich beeinträchtigt. Aber auch die Verarbeitung von Stahlbeton führt zu einer nachteiligen Charakteristik. Im Wesentlichen störungsfrei erfolgt die Kommunikation in freier Natur oder auch bei der Durchquerung von Holzwänden oder Glas. Eine Störquelle der besonderen Art kann ein Mikrowellengerät darstellen. Da dieses auf der gleichen Frequenz »funk« und zudem eine – trotz Abschirmung – oft mehr als hundertfache Abstrahlung produziert, sollte ein WLAN Access Point oder WLAN-Controller in möglichst ausreichender Entfernung zum Mikrowellengerät positioniert werden.

Unter Hotspots versteht man öffentliche WLAN-Zugangspunkte, die an Flughäfen, in Restaurants, Krankenhäusern oder Hotels den Gästen bzw. Patienten zur Verfügung gestellt werden. Die Abrechnung für die Dauer der Nutzung erfolgt entweder über bereits vorhandene Identifikationskriterien (Benutzername, Kennwort) bei sogenannten WISPs (*Wireless Internet Service Provider*) oder auch anderen Providern, bei denen der Kunde bereits ein Abrechnungskonto besitzt (z.B. Mobilfunk-Provider). In einigen Fällen kann die Nutzung auch aufgrund käuflich erworbener Zeitkontingente erfolgen.

Sicherheitsaspekte

Insbesondere bei der Nutzung von Hotspots ist die Sicherheit der Datenkommunikation von großer Bedeutung. Funknetze bieten wegen der recht einfachen Zugangsmöglichkeit eine besonders gute Angriffsfläche für Hacker bzw. Cracker, sofern keine ausreichenden Sicherheitsmaßnahmen getroffen wurden. Die folgende Aufstellung zeigt eine Übersicht der wichtigsten Kriterien, die für die Realisierung einer minimalen Basissicherheit in WLANs berücksichtigt werden sollten:

- *ESSID*

Die ESSID (*Extended Service Set Identifier*) ermöglicht die Zuordnung einer eindeutigen ID. Bei vielen Produkten wird als Standardwert für die ESSID der Modellname des WLAN-Routers verwendet. Wird hier keine eigene Kennung der WLAN-Zelle vergeben, ist die Wahrscheinlichkeit unliebsamer »Zuhörer« während der eigenen Kommunikation recht hoch. Da diese Kennung bei Hotspots in der Regel allen Benutzern bekannt gemacht wird, ist die Sicherheitsqualität der ESSID eher fragwürdig.

- *MAC-Adresse*

Jeder WLAN-Controller besitzt – wie übrigens jede andere Netzwerkkomponente auch – eine weltweit eindeutige *burnt-in address*. Diese Adresse (MAC = *Media Access Control*) sollte

bei der Konfiguration des Access Point dediziert angegeben werden, sodass ein Zugriff von Benutzern mit anderen MAC-Adressen zurückgewiesen wird.

- *DHCP*

Die Aktivierung der dynamischen Vergabe von IP-Adressen mittels DHCP (*Dynamic Host Configuration Protocol*) ist in einem WLAN nicht zu empfehlen. Denn sollte es jemandem gelingen, in das eigene Netzwerk einzudringen, würde er automatisch eine IP-Adresse erhalten, mit der er sich dann im Netzwerk frei bewegen kann. Eine dedizierte Vergabe von IP-Adressen erhöht in kleineren Netzen den Administrationsaufwand nur unerheblich, stellt aber einen weiteren wirkungsvollen Schutz gegenüber Eindringlingen und ihren Aktivitäten dar.

- *Verschlüsselung*

Zur Wahrung der Datenintegrität sollten Schlüssel zur Datenverschlüsselung angelegt werden. Die hierbei anfänglich verwendete Technik wird als WEP (*Wired Equivalent Privacy*) bezeichnet und heute als nicht mehr ausreichend sicher angesehen. Das aktuell empfohlene Verschlüsselungsverfahren WPA3 wird bei Wi-Fi-6-basierter Funk-Hardware meist direkt implementiert und sollte mit eigenen Schlüsseln versehen werden. Diese sollten allerdings aus Sicherheitsgründen regelmäßig geändert werden.

HINWEIS

Bei der Festlegung der WPA-Version im jeweiligen Gerät ist darauf zu achten, dass die gewählte Version mit allen teilnehmenden Komponenten kompatibel ist. Können in einigen Netzknoten lediglich WPA2-Schlüssel eingerichtet werden, so sind die entsprechenden Geräte nicht in der Lage, mit WPA3-abgesicherten Komponenten zu kommunizieren. Als Konsequenz muss dann – je nach Sicherheitsanforderung – entschieden werden, ob die ältere, lediglich WPA2-fähige Hardware ausgetauscht werden muss oder ob ein »Downgrading« auf WPA2 als ausreichend angesehen wird.

Die *Wired Equivalent Privacy* bezeichnet ein auf dem RC4-Algorithmus (*Rivest Cypher No. 4*) beruhendes Verschlüsselungsverfahren, das von einem der weltweit bekanntesten Kryptografen, Ron Rivest, bereits 1987 entwickelt wurde. WEP verwendet Schlüssellängen von 40 und 104 Bit (die ebenfalls in diesem Zusammenhang oft genannten 64- und 128-Bit-Verschlüsselungen beziehen sich lediglich auf zusätzliche Initialisierungsvektoren von 24 Bit Länge, die allerdings aus Sicherheitsgründen wegen ihrer Angreifbarkeit vermieden werden sollten).

Der Standard IEEE 802.11i stellt eine Erweiterung der WLAN-Standards um einige wichtige Sicherheitsfunktionen dar, die WEP nicht liefern kann. So ist beispielsweise der Einsatz des DES-Nachfolgealgorithmus AES (*Advanced Encryption Standard*) vorgesehen, der auf einem 128-Bit-Blockchiffrieralgorithmus beruht und sogar auf 256 Bit erweiterbar ist. Eine weitere sinnvolle Option zur Erhöhung der Datensicherheit stellt die Verwendung des IP-Sicherheitsprotokolls IPsec im Rahmen von VPNs (*Virtual Private Networks*) dar.

Basierend auf dem Standard 802.11i wurde das WPA (*Wifi Protected Access*) von der Wi-Fi-Alliance ausgegliedert, das seit Januar 2018 in der Version 3 (WPA3) vorliegt. Ihm liegen ein 192-Bit-Schlüssel im Enterprise-Modus und ein 128-Bit-AES-Schlüssel im Personal-Modus zugrunde.

Von den möglichen Sicherheitsvorkehrungen und technischen Möglichkeiten abgesehen, sollten alle Zugriffe auf die Konfiguration eines Access Point über gesonderte Kennwörter abgesichert sein.

Bluetooth

Mit der Bluetooth-Technologie wurde, im Vergleich zum WLAN, ein ganz anderer Ansatz bzw. ein anderes Ziel verfolgt. Auch wenn es technisch im selben 2,4-GHz-Band wie ein WLAN

eingesetzt wird, so ist seine Funktionalität darauf ausgelegt, über relativ kurze Entfernungen Endgeräte unterschiedlichsten Charakters miteinander kommunizieren zu lassen. Der Betrieb eines infrastrukturellen Netzwerks stand bei Bluetooth nie im Vordergrund; vielmehr sind typische Komponenten einer Bluetooth-Kommunikation Mobilfunkgeräte wie Smartphones, Smartwatches oder Peripheriegeräte (z.B. Headsets) und sicher vereinzelt auch PCs, Notebooks oder Tablets.

Bluetooth wird daher vorwiegend in sogenannten *Wireless Personal Area Networks* (WPAN) eingesetzt, einem Verbund einzelner Endgeräte, der in der Vergangenheit sehr oft der Infrarot-Technologie vorbehalten war (IrDA).

Wie bereits in [Abschnitt 1.2.2](#) ersichtlich, gibt es heute seit der Bluetooth-Spezifikation 5 (Einführung im Dezember 2016) bzw. in der zuletzt aktualisierten Version 5.3 (Einführung im Juli 2021) durchaus überlappende Einsatzbereiche mit IEEE-802.11-WLAN-Technologien, insbesondere hinsichtlich Low-Energy-Einsatz. Bluetooth kann zwar bei einer deutlich reduzierten Datenrate von unterhalb 1 Mbit/s innerhalb von Räumen und Gebäuden eine Reichweite von 40 Metern und im Freigelände bis zu 200 Metern erreichen. Damit stellt es allerdings keine direkte Konkurrenz von Hochgeschwindigkeits-Funknetzen im IEEE-802.11-Standard

dar. Nicht zuletzt deshalb, weil dies der Bluetooth-Standard auch überhaupt nicht anstrebt. Sein Schwerpunkt liegt wie bereits dargestellt im Nahbereich und in der Kommunikation mobiler Endgeräte.

Audio-Streaming und Standort-Bestimmung (Lokalisierung von Personen und Gegenständen) sind weitere Disziplinen, mit denen sich der Standard seit einigen Jahren beschäftigt.

Die Bluetooth Special Interest Group ist als Non-Profit-Organisation für die Entwicklung des Standards verantwortlich. Ihr hat sich eine Vielzahl von Unternehmen angeschlossen, die an der Weiterentwicklung von Bluetooth mitarbeiten.

Sonstige Varianten

Die im Folgenden kurz dargestellten Netzwerktopologien besaßen in der Vergangenheit große Relevanz, sind aber heute zunehmend den zuvor beschriebenen Technologien gewichen. In einzelnen Unternehmen und öffentlichen Infrastrukturen kommen sie zwar immer noch vor, allerdings liegt das zumeist an den hohen Kosten, die für den Austausch von Netzwerken zu berücksichtigen sind. Mit den realisierbaren Geschwindigkeiten beim Datentransport sind sie einfach nicht mehr konkurrenzfähig.

Token Ring

Mitte der 80er Jahre des vorigen Jahrhunderts verfeinerte die Firma IBM für ihre eigenen Erfordernisse ein Netzwerkverfahren, das bereits 1972 entwickelt, aber erst 1989 unter dem Namen *Token Ring* auf den Markt gebracht wurde. Bei Token Ring können neben den alten starren, mehrfach verdrehten Vierdrahtkabeln (Typ-1-Kabel) auch neuere Kabeltypen (z.B. Cat5, Cat6 oder Cat7 mit RJ45-Steckern) verwendet werden. In Abhängigkeit vom Zugriffsverfahren lassen sich Kapazitäten bis zu 20–30 Mbit/s erreichen. Begann man zunächst beim Token Ring mit einer LAN-Geschwindigkeit von 4 Mbit/s, so wurde seit 1990 immer mehr die wesentlich leistungsfähigere 16-Mbit/s-Variante eingesetzt, insbesondere in LAN-übergreifenden Backbones.

Prinzip und Zugriffsverfahren

Im Token Ring kommt mit *Token Passing* ein völlig anderes Netzwerkzugriffsverfahren als im CSMA/CD-Verfahren von Ethernet zur Anwendung; der entsprechende Standard ist als IEEE 802.5 definiert. Auch hier erfolgt die Anschaltung an das Netzkabel über einen Controller (für PCs, Workstations u.Ä.). Allerdings geht ein solcher Controller (bzw. Rechner) nicht direkt an den Ring, sondern wird normalerweise zunächst

zu einem Ringleitungsverteiler geleitet, an dem weitere sieben Stationen gebündelt werden (die Anzahl der gebündelten Stationen hängt vom Typ des Ringleitungsverteilers bzw. von der Anschlusstechnik ab). Dieser besitzt für jede Station, die den Ring betreten möchte, ein passiv oder aktiv arbeitendes Relais. Die Stromversorgung der einzelnen Relais wird über den Controller der Station übernommen. Ist die Station ausgeschaltet, wird der Netzbetrieb an diesem (nun stromlosen) Relais vorbeigeführt. Erst nach Einschalten des Rechners und Aktivierung des Controllers (Treibersoftware) erhält die Station über das nun geöffnete Relais einen Netzwerkzugang. Diese kombinierte Ring-Stern-Topologie ist daher in diesem Verfahren relativ unempfindlich gegen Ausfälle einzelner Stationen.

Die erste Station, die den Ring betritt, erhält die Funktion des aktiven Monitors. Sie generiert ein sogenanntes *Frei-Token*, die Sendeberechtigung, und sorgt ferner für seine Überwachung. Alle weiteren Stationen werden als »Stand-by-Monitore« geführt. Sie übernehmen erst dann aktive Monitorfunktionen, wenn der bislang aktive Monitor ausfällt.

Bei einem Token (hier: Frei-Token) handelt es sich um ein 3-Byte-Bitmuster, bestehend aus *Starting Delimiter*, *Access Control* und *Ending Delimiter*. Indem die gewünschten Netzwerkdaten

(eigene Adresse, Adresse der Zielstation und Nutzdaten) angehängt werden, entsteht so der Token-Ring-Frame, der von Station zu Station weitergereicht wird, bis er die angesprochene Zielstation erreicht. Nachdem der Frame kopiert und entsprechend markiert wurde, wird er wieder auf den Ring gebracht und nach gleichem Verfahren so lange weitergereicht, bis er an der Sendestation wieder angekommen ist. Diese nimmt ihn schließlich vom Netz und generiert ein Frei-Token, das dann von anderen sendewilligen Stationen für eine Übermittlung von Daten verwendet wird. Hier ist es wichtig festzuhalten, dass eine Station lediglich dann senden kann, wenn sie im Besitz eines Frei-Tokens ist.

Das Token-Ring-Verfahren geht von genau einem Token pro Sendeprozess im Ring aus. Bei einer Geschwindigkeit von 4 Mbit/s ist diese Begrenzung auch unbedingt sinnvoll, da die Speicherfähigkeit des Rings, der normalerweise einige Hundert bis wenige Tausend Meter Länge (Kabel) nicht übersteigt, einfach nicht ausreicht. Als Faustformel wird bei 4-Mbit/s-Ringen von 50 Meter langen Bits gesprochen (in Abhängigkeit von der Datenrate des Protokolls, der Ringlänge in Metern und der Signalausbreitungsgeschwindigkeit = etwa halbe Lichtgeschwindigkeit). Ein 1000 Meter langer Ring wäre demnach schon bei einem gut zwei Byte langen Token samt Daten vollständig ausgefüllt. Eine sendende Station wird daher

schon während ihres laufenden Sendeprozesses ihre über den Ring weitergereichte Nachricht wieder empfangen.

Bei einer Geschwindigkeit von 16 Mbit/s erhöht sich zwar die Ringkapazität um das Vierfache (d.h. etwa 8 Bytes für die Belegung des 1000 Meter langen Ringes), eine wesentliche Veränderung des Verhaltens im Ring wird dadurch jedoch nicht erreicht. Das wird drastisch anders, wenn Backbones von mehreren Kilometern Länge zum Einsatz kommen. Die Ring-Speicherfähigkeit nimmt also deutlich zu, sodass auch mehrere Frames (aber immer nur ein einziges Token) zu einem bestimmten Zeitpunkt im Ring verweilen. Dieses Verfahren wird *Early Token Release* genannt.

Das nach IEEE 802.4 standardisierte Token-Bus-Verfahren kommt hauptsächlich in den USA zur Anwendung (ARCNET entspricht zwar nicht diesem Standard, verwendet aber ein sehr ähnliches Verfahren). Unter Ausnutzung der Stärken und Vermeidung von Schwächen der jeweiligen Konzeption stellt es eine Kombination der Bus- und Ringtechnologie dar. Auf dem physikalischen Bus wird ein logischer Ring implementiert.

High Speed Token Ring (HSTR)

Namhafte Hersteller von Token-Ring-Produkten versuchten Ende des vorigen Jahrhunderts (1997), dem gestiegenen Bedarf

an Hochgeschwindigkeitsnetzen auch für Token Ring Rechnung zu tragen. Gegenüber dem bereits etablierten Fast Ethernet konnte die Token-Ring-Gemeinde noch nichts Gleichwertiges vorweisen. So kam es insbesondere durch Mitwirkung der Hersteller IBM, CISCO, 3COM, Bay Networks und Madge Networks zur Standardisierung des *High Speed Token Ring* als spezielle Norm IEEE 802.5u.

Unter Beibehaltung des bekannten Token-Ring-LAN-Zugriffsverfahrens, des *Source Route Bridging*, und der gegenüber Ethernet größeren Datenpakete ist der HSTR jedoch primär zur Ablösung bestehender Backbone-Konzepte gedacht, die bislang mit 16 Mbit/s betrieben wurden, oder aber zur Einrichtung dedizierter Punkt-zu-Punkt-Serververbindungen. Der Betrieb auf einem Medium, an dem mehrere Stationen bedient werden müssen, war prinzipiell nicht vorgesehen.

Sehr schnell wurde klar, dass die Token-Ring-Technologie samt HSTR mittelfristig keine große Rolle mehr spielen würde. Fast alle namhaften Hersteller von Token-Ring-Produkten haben sich mittlerweile von dieser Technologie getrennt. Selbst IBM, Stammvater des Token Ring, hat sich 1998 aus dem Token-Ring-Geschäft zurückgezogen.

FDDI

Das *Fiber Distributed Data Interface Network* beschreibt die konsequente Weiterentwicklung des IEEE-802.5-Standards unter ISO/ANSI X3T9.5 auf Basis eines Doppelrings (Primär- und Sekundärring). Es ist für hohe Geschwindigkeiten von 100 Mbit/s ausgelegt und damit für komplexe Backbone-Konzepte geeignet. Der Backbone sollte eine maximale Netzausdehnung von etwa 100 bis 200 km nicht übersteigen und seine angeschalteten (max. 500 bis 1000) Stationen sollten nicht weiter als 2 km voneinander entfernt sein.

Im ISO/OSI-Referenzmodell wird für die FDDI-Technologie der erste Layer nochmals in zwei Sub-Layer unterteilt. Der unterste, auch *PMD-Sub-Layer* genannt (PMD = *Physical Medium Dependent*), beinhaltet Definitionen für den Zugriff auf das GlasfasermEDIUM, der obere PHY-Sub-Layer (PHY = *Physical Layer Protocol*) sorgt für die medienunabhängige Codierung der verwendeten Signale.

Ähnlich wie beim Early-Token-Release-Verfahren können sich zu einem bestimmten Zeitpunkt im FDDI-Ring mehrere Nachrichten gleichzeitig befinden. Eine sendende Station im Ring generiert nach Abgabe ihrer Nachricht an das Netz ein Frei-Token. Dieses kann eine weitere, sendewillige Station

benutzen, die dann ihrerseits die gewünschten Daten über das Netzwerk überträgt. Infolge der z.T. gewaltigen Ausdehnung eines FDDI-Backbones gibt es daher bei mehreren kursierenden Nachrichten im Netz keinerlei Kapazitätsprobleme.

Man unterscheidet zwei Typen von Stationen: Class-A-Stationen sind sowohl an den Primär- als auch an den Sekundärring angeschlossen, Class-B-Stationen allerdings nur an den Primärring. Die Anbindung der Class-B-Stationen erfolgt über sogenannte Konzentratoren (*Wiring Concentrator*). Bei einem Leitungsausfall innerhalb eines Rings kann eine A-Station durch Umschalten auf den anderen Ring ein Backup herstellen. Dieser Vorgang erfolgt zumeist unbemerkt. Eine Kommunikationsunterbrechung wird nicht hervorgerufen.

FDDI ereilte ein ähnliches »Schicksal« wie Token Ring, da mittlerweile auch für hohe Übertragungsgeschwindigkeiten im LAN fast ausschließlich Ethernet-Technologien eingesetzt werden.

xDSL

Ein konkretes Beispiel für Breitband-ISDN sind die xDSL-Dienste (ISDN selbst wird mittlerweile von fast allen Providern weltweit nicht mehr betrieben). Hier wird unterschieden nach Übertragungsmedium (Funk, Stromleitung, Kupfer und

Lichtwellenleiter), Charakteristik (Simplex, Halbduplex, Duplex) und Datenstrom (symmetrisch oder asymmetrisch). Beim asymmetrischen Duplexverfahren über Kupfer spricht man von ADSL, der vor einigen Jahren verbreitetsten Variante. Dabei differieren die Parameter für Downstream-Datenübertragungen zum lokalen Rechner und Upstream-Übertragungen zum Service-Provider. Es wurden seinerzeit unterschiedliche Bandbreiten angeboten, die von 768 kbit/s bis zu 1000 Mbit/s reichten; aufgrund der teilweise schlechten Leitungsqualität der Kupferkabel gab es jedoch Einschränkungen in Bezug auf die theoretischen Maximalwerte.

ADSL war in diesem Zusammenhang sehr interessant, da es sich dabei primär um eine Download-Charakteristik handelte, die ein Vielfaches der in der Vergangenheit möglichen 64-kbit/s-Bandbreite zur Verfügung stellen konnte. In Verbindung mit den üblichen Flatrates stellte ADSL eine preiswerte Lösung für eine schnelle Internetkommunikation dar. Für diesen Einsatz wurde ein neuer Standard konzipiert: ADSL2 bzw. ADSL2+. Diese von der International Telecommunication Union (ITU) verabschiedeten Standards stellten Bandbreiten nahezu beliebiger Größe zur Verfügung – einschließlich einer Reihe technischer Verbesserungen hinsichtlich Reichweite und Fehlerdiagnose.

Mit einer Weiterentwicklung namens *DSL-Vectoring* geht eine Ausdehnung der Übertragungsgeschwindigkeiten bis zu (praktisch verfügbaren) 100 Mbit/s einher. Zur Steigerung der Übertragungsgeschwindigkeit erfolgt dabei ein Ausgleich der elektromagnetischen Störstrahlungen (Übersprechen) zwischen den Leitungen (DSL-Kupferdoppelkabel). Durch das Vectoring werden die Störstrahlen minimiert und damit die verfügbare Kapazität erhöht.

ATM

Das ATM-Verfahren verwendet für den Datentransport ATM-Zellen, die allesamt die gleiche Länge von 53 Bytes (48 Bytes Daten, 5 Bytes Header) besitzen. Die daraus zusammengesetzten Bit-Datenströme können innerhalb eines ATM-Netzwerks unterschiedlich sein. Die Parallelität von Bit-Strom A mit einer Kapazität von 2 Mbit/s und Bit-Strom B mit 64 kbit/s, aber auch Bit-Strom C mit 10 Mbit/s ist möglich. Bit-Ströme lassen sich als *virtuelle Kanäle* auch zu *virtuellen Pfaden* zusammenfassen, wobei jeder virtuelle Pfad die Verbindung zwischen zwei Endeinrichtungen vornimmt. Die eindeutige Identifizierung innerhalb der virtuellen Pfade und Kanäle erfolgt durch das VCI-VPI-Paar (*Virtual Channel Identifier*; *Virtual Path Identifier*) im 5-Byte-Header als Bestandteil der ATM-Zelle.

In einem asynchronen Zeit-Multiplexing werden die ATM-Zellen auf das Medium gebracht, d.h., sie belegen asynchron freie Zeitschlitz und werden innerhalb des Übertragungskanal nicht fest zugeordnet. Die durch kleine Zellen (kurze Verzögerungszeiten) und asynchrones Zeit-Multiplexing (Weitergabe von Zellen unabhängig von definierten Zeittakten) erreichbaren Geschwindigkeiten liegen bei über 600 Mbit/s (ATM 622).

Ein ATM-Netzwerk besteht aus einer Sammlung sog. *ATM-Switches*, die zu beliebig großen Netzwerken ausgebaut werden können. Jeder Switch hat üblicherweise jeweils zwei Ein- und Ausgänge. Die Switching-Funktion wird durch eine Verbindung eines Eingangs mit seinem diagonal angeordneten Ausgang hergestellt. Dabei beruht die eigentliche Leistungsfähigkeit des ATM auf den kleinen Zellen mit fester Länge, dem asynchronen Zeit-Multiplexing und den leistungsfähigen Prozessoren der ATM-Switches.

Aufgrund der rasanten Weiterentwicklung im Ethernet-Bereich hat die Verbreitung von ATM nun auch ein Ende gefunden, da Ethernet auch mehr und mehr den WAN-Bereich – also das klassische Einsatzgebiet von ATM – eroberte.

Netzwerkkomponenten

Netzwerke sind mittlerweile ein fester Bestandteil innerbetrieblicher Kommunikationsinfrastrukturen, auf die niemand mehr verzichten kann oder möchte. Dabei wird je nach Größe, Ausdehnung und Komplexität eines Netzwerks neben der Verkabelung und der entsprechenden Anschlusstechnik auch eine Vielzahl weiterer Netzwerkkomponenten benötigt. So wie die Infrastruktur eines Netzwerks mit der Verkabelung und der Anschlusstechnik als der passive Teil eines Netzwerks bezeichnet wird, gehören andere Bestandteile bzw. zusätzliche Hardwarekomponenten eines Netzwerks zum aktiven Teil. Insbesondere gehören zu den aktiven Komponenten Bauteile wie Switches, Hubs, Repeater, Brücken oder Router.

Repeater

Zur einfachen Verbindung zweier Netzwerk-Segmente (Ethernet oder auch WLAN) auf physikalischer Ebene werden Repeater eingesetzt. Dabei handelt es sich um ein Gerät, das elektrische Signale entgegennimmt, diese verstärkt und anschließend weiterleitet. Bei einigen Modellen erfolgt eine Bereinigung des Signals von Störgeräuschen, bevor es nach einer erneuten Codierung weitergeleitet wird. Meistens wird

jedoch keinerlei Intelligenz zur Fehlerbehandlung implementiert, sondern die Signale werden einfach nach Verstärkung durchgereicht. Solche Konfigurationen kommen dann zum Einsatz, wenn die Signalstärke bei längeren Strecken abnimmt und somit – ohne Repeater – am Ziel keine ausreichende Signalqualität mehr besitzen.

Brücke

Sobald zwei oder mehr Netzwerke gleichen oder verschiedenen Typs miteinander zu einem logischen Netzwerk verbunden werden, sind dafür entsprechende Koppellemente vonnöten. Ein solches Koppellement ist beispielsweise eine *Brücke* (*Bridge*), siehe auch [Abbildung 1-5](#). Dabei kann es sich um Rechner handeln, die mit Einsatz bestimmter Netzwerksoftware Datenpakete von einem Netzwerk in das andere leiten.

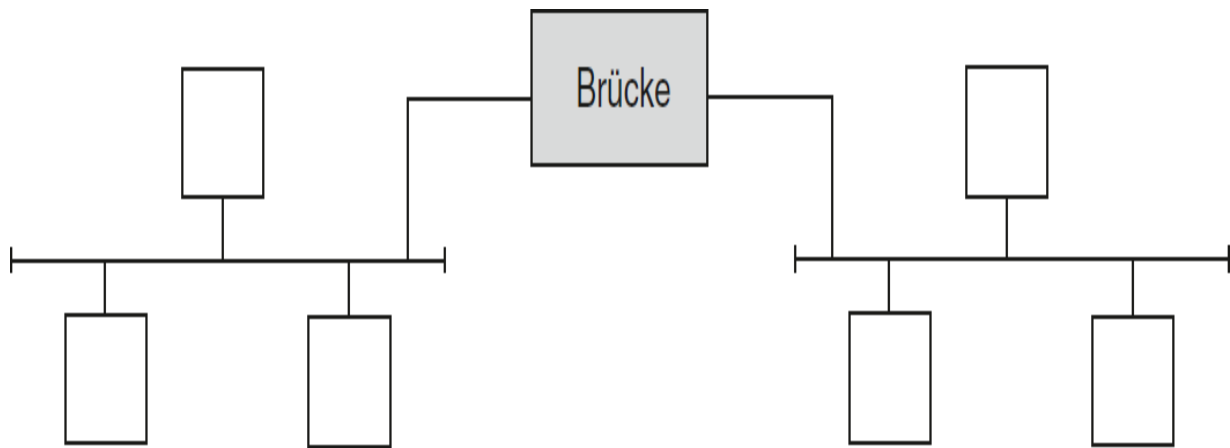


Abb. 1–5 *Eine Brücke verbindet zwei unterschiedliche Netzwerkbereiche miteinander.*

Weit häufiger werden allerdings dedizierte Brücken als separate Netzwerkkomponenten eingesetzt, die von verschiedenen Herstellern angeboten werden.

Mit dem einfachen Kopieren und Weiterleiten von Datenpaketen ist es allerdings nicht getan. Brücken sind als Koppellemente innerhalb von Netzwerken auch für eine Abschottung zu benachbarten LAN-Segmenten im Falle von Hardwarefehlern zuständig. Sie treffen darüber hinaus anhand erlernter Adresstabellen die Entscheidung, ob ein Datenpaket weitergeleitet werden muss oder nicht. Im letzteren Fall handelt es sich dann um lokalen Datenverkehr, der das eigene LAN-Segment nicht verlassen darf. Segmentübergreifender Datenverkehr wird von der Brücke über die eigene

Netzwerkgrenze hinweggeführt und ins benachbarte LAN-Segment übertragen.

Das Bridging zwischen zwei Netzwerken gleicher oder unterschiedlicher LAN-Zugriffsverfahren findet auf ISO/OSI-Schicht 2 (Layer 2), der Datumsicherungsschicht, statt. Alle höheren Protokolle verhalten sich bezogen auf das Bridging transparent, d.h. werden nicht interpretiert, sondern lediglich weitergereicht. Somit kann auf den getrennten Netzwerken ein Protokoll wie TCP/IP eingesetzt werden, ohne dass Netzwerkgrenzen erkennbar werden.

HINWEIS

Brücken werden nicht ausschließlich dazu eingesetzt, Netzwerke zu *verbinden*. Es ist oft erwünscht, dass sie eine *trennende* Wirkung haben. Dies ist beispielsweise dann der Fall, wenn Brücken als Instrumente zur Netzwerkstrukturierung eingesetzt werden, wobei die Funktion einer Brücke heutzutage in der Regel in einem Switch (siehe dort) enthalten ist.

Eine nachvollziehbare *Fehlereingrenzung* im LAN kann nur dann gewährleistet werden, wenn es in sauber strukturierte, überschaubare Segmente unterteilt ist. LAN-Segmente, die eine

technisch maximal mögliche Anzahl von Stationen beherbergen, gehören aus Geschwindigkeitsgründen der Vergangenheit an. Besonders stark frequentierte Server werden zunehmend über *Switches* angebunden. Diese Maschinen teilen sich dann das Medium nicht mit anderen Netzwerkkomponenten, sondern benutzen es allein in einer Punkt-zu-Punkt-Verbindung und können daher mit einer großen Netzwerkbandbreite arbeiten.

Brücken sollen auch dafür sorgen, dass überall dort, wo Datenpakete eines bestimmten Protokolls unerwünscht sind, diese herausgefiltert werden und somit die Netzwerkgeschwindigkeit nicht unnötig negativ beeinflussen. Zur Installation einer dynamischen Ausfallsicherheit des Netzwerks werden Brücken oft redundant eingesetzt, d.h., es wird eine parallele Installation vorgesehen, wobei eine der beiden Brücken allerdings keine aktive Brückenfunktionalität übernimmt, sondern im Hot-Stand-by-Modus gefahren wird. Sobald eine der beiden Brücken ausfällt, springt die jeweils andere ein und übernimmt die volle Funktionalität der defekten Brücke. Die dabei entstehende Problematik der *Zyklusbildung* im Ethernet-LAN wird durch den *Spanning-Tree-Algorithmus* neutralisiert.

Funktionsweise

Brücken beziehen ihre Topologie-Informationen aus einem Tabellenwerk, das sie selbst generieren (Lernprozess) und verwalten müssen. Wenn eine Station ins Netzwerk genommen wird, so meldet sie sich zunächst bei ihrer Brücke mit einem: »Hallo, hier bin ich!« Damit die Brücke auch erfährt, wer sie gerade so freundlich begrüßt, fügt die Station ihrer Begrüßung auch gleich ihren Namen hinzu. Dieser Name wird durch eine eindeutige MAC-Adresse repräsentiert (in [Abb. 1–6](#) vereinfacht als zweistelliger Zahlenwert dargestellt). Alle Stationen, die sich bei der Brücke auf diese Art und Weise gemeldet haben, werden von ihr gelernt. Dieser Lernprozess bezieht sich auf beide (bzw. alle) LAN-Segmente, die durch die Brücke verbunden werden. Allerdings merkt sich die Brücke, von wo die Begrüßung an sie gerichtet wurde, d.h. über welchen Port. Mit diesem Wissen kann sie jederzeit Aufträge zur Versendung von Datenpaketen bearbeiten. Lokale Datenpakete werden nicht übertragen, Datenpakete, die an Stationen des Nachbarnetzes gerichtet sind, werden durchgeschleust.

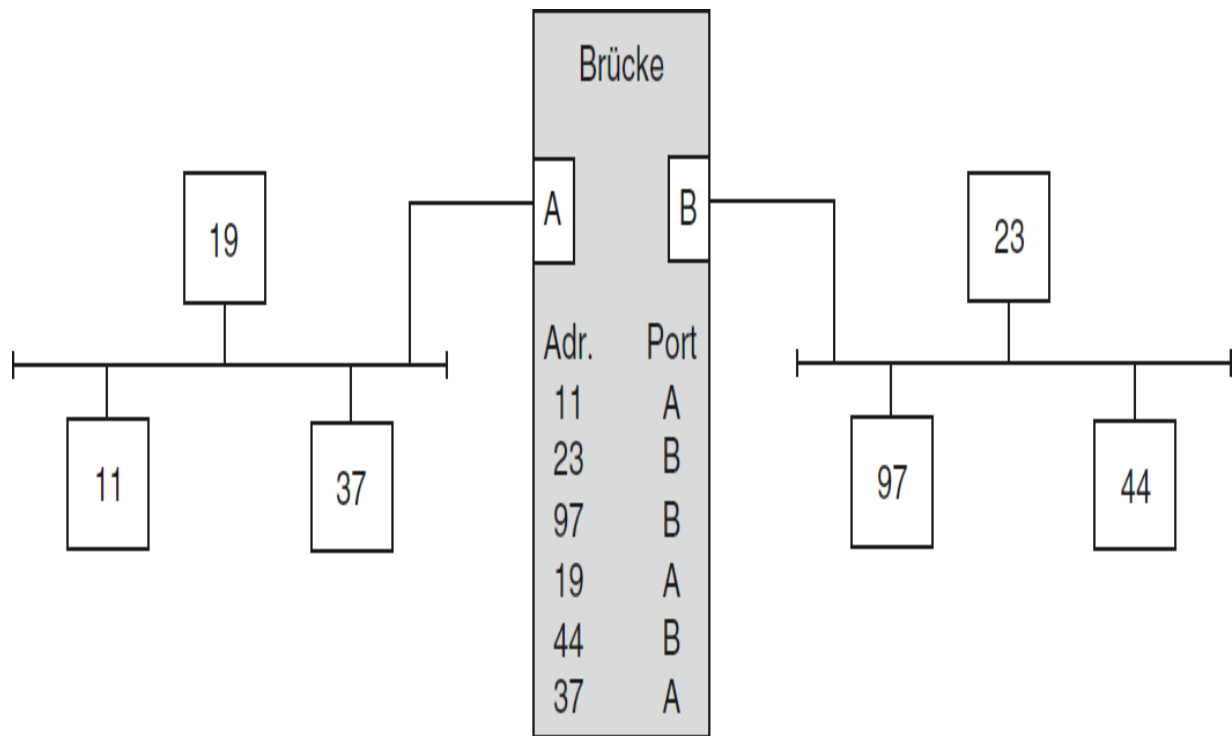


Abb. 1–6 *Erlernte Adresstabelle einer Brücke*

In definierbaren Zeitintervallen werden alte Adresseinträge in den Tabellen gelöscht, um ein Aufblähen der Adressinformationen zu vermeiden und einen jederzeit aktuellen Tabellenstand zu gewährleisten. Dieses zeitgesteuerte dynamische Austauschverhalten wird auch als *Aging* bezeichnet. Die Notwendigkeit eines solchen Verfahrens leuchtet unmittelbar ein, wenn man an den Standortwechsel von Stationen denkt. Gelernte Topologie-Informationen stimmen dann nicht mehr (bzw. sind nicht eindeutig), und die deaktivierte Station kann nicht mehr angesprochen werden.

Bridging-Verfahren

Beim Betrieb eines größeren Netzwerks, das sich aus kleineren Netzsegmenten zusammensetzt, die durch selbstlernende Brücken untereinander verbunden sind, kann es bei redundanten Strukturen (parallelen Brücken) zum endlosen Zirkulieren von Datenpaketen kommen. Durch ein Fehlen von RIF-Feldern (RIF = *Routing Information Field*), die eine Limitierung der Brückenpassagen vornehmen, kann eine Endlosschleife nicht verhindert werden. Die einzelnen Brücken wissen ja nicht, ob sie der Frame schon einmal passiert hat. Diesem Effekt wirkt ein Verfahren zur Unterdrückung von Endlosschleifen entgegen, das von der Firma Digital für Ethernet-LANs entwickelt wurde: *Spanning Tree*. Da bei diesem Verfahren die einzelnen Stationen nicht beteiligt sind, spricht man auch vom *Transparent Bridging* (TB).

Die Umsetzung dieses schleifenunterdrückenden Algorithmus erfolgt dadurch, dass überall dort, wo innerhalb eines Netzwerkkomplexes potenzielle Zyklen entstehen, die entsprechenden redundanten Verbindungswege derart deaktiviert werden, dass eine vermaschte Baumstruktur erzeugt wird. In dieser Baumstruktur existiert immer nur genau eine Route zwischen zwei Stationen. Die Deaktivierung bestimmter Verbindungswege bzw. ihrer Brücken wird

allerdings dann wieder aufgehoben, wenn der »Spanning Tree« durch eine Störung unterbrochen ist.

In der Praxis wird dieses Verfahren über eine Prioritätensteuerung realisiert. Netzwerkadapter bzw. Leitungen können außerdem mit einem Kostenfaktor bewertet werden, der schließlich zur Routenbildung führt. Sind die kumulierten Kosten für eine Route gleich, so entscheiden die den Brücken zugeordneten Prioritäten. Ferner müssen Ports und Brücken eindeutig adressierbar sein. Eine kontinuierliche Kommunikation der Brücken untereinander (*Bridge Protocol Data Units* = BPDU) gewährleistet ein schnelles Backup, wenn Teilbereiche eines Netzes ausfallen sollten. Wird eine weitere Brücke ins LAN gebracht, so muss natürlich der Spanning Tree neu berechnet werden. Für diese Zeit (einige Sekunden bis wenige Minuten) ist im LAN keine Kommunikation möglich: Das Netzwerk »steht«.

HINWEIS

Durch die ständige Weiterentwicklung der Ethernet-Technologien (Fast Ethernet, Gigabit Ethernet) hat das Bridging-Verfahren mehr und mehr an Bedeutung verloren

bzw. wird diese Funktion in der Switch-Technologie abgebildet (siehe dort).

Switch

LAN-Switche erfreuen sich seit Beginn dieses Jahrhunderts einer stetig steigenden Beliebtheit und haben klassische Konzepte auf Basis von Brücken (*Bridges*) bereits weitgehend abgelöst – dies sicherlich nicht zuletzt deswegen, weil sich die Grundkonzepte sehr ähneln. Bei Switches handelt es sich, wie bei den Brücken, um Geräte des Layers 2 (siehe [Abb. 1–7](#)), die von zahlreichen Herstellern angeboten werden. Eine funktionale Abgrenzung zu Brücken ist oft nur in Verbindung mit den jeweils konkreten Produkten der einzelnen Hersteller möglich, da diese sehr unterschiedliche Leistungsmerkmale implementieren.

Genau wie Brücken stellen auch LAN-Switches Koppellemente dar, die insbesondere eine Anbindung einzelner Netzwerkkomponenten (z.B. individuelle Server, aber auch Workstations) ermöglichen. Außerdem können sie diesen Komponenten eine garantierte Bandbreite über einen direkten Portanschluss zur Verfügung stellen. LAN-Switches werden normalerweise mit mehreren Ports (von wenigen bis weit über

hundert Ports) im LAN installiert und wie Multiport-Brücken behandelt.

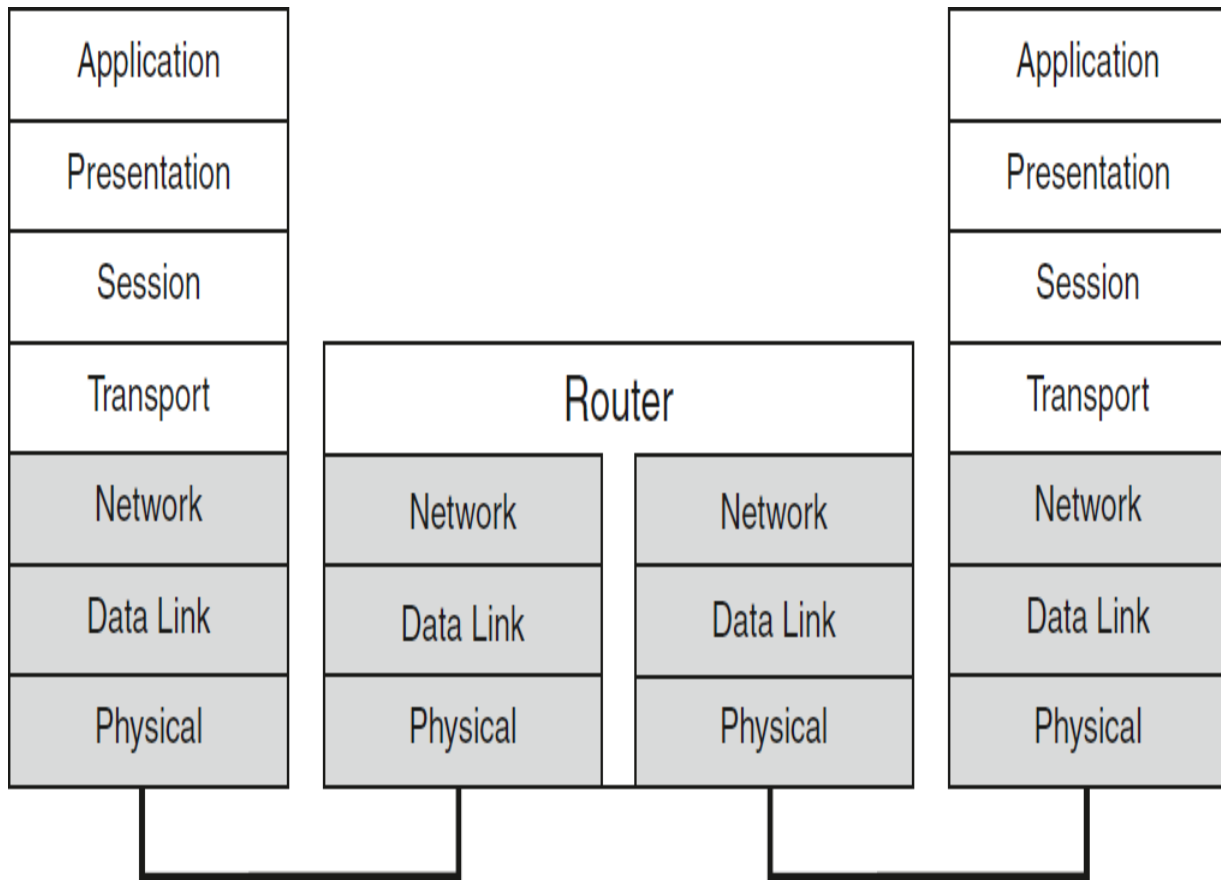


Abb. 1–7 Bridge bzw. Switch auf der zweiten OSI-Schicht

HINWEIS

Auch wenn das klassische Switching im Layer 2 beheimatet ist, so werden mittlerweile Systeme eingesetzt, die im ISO/OSI-Layer 3 (Network) und sogar 4 (Transport) und darüber ihre Dienste verrichten.

Das Switching ist die konsequente Weiterentwicklung des Brückenkonzepts (siehe vorhergehender Abschnitt), das auf gemeinsam genutzten Medien (*Shared Media*) eine segmentübergreifende Kommunikation ermöglicht. Diese Weiterentwicklung wurde notwendig, weil ein performanter Betrieb zentraler Dienste (Datenbankserver, Internetdienste, Intranetserver, Mail-Server usw.) im Netzwerk zur Verfügung gestellt werden muss und dies bei Einsatz von *Shared Media* nicht gewährleistet werden kann (siehe [Abb. 1–8](#)).

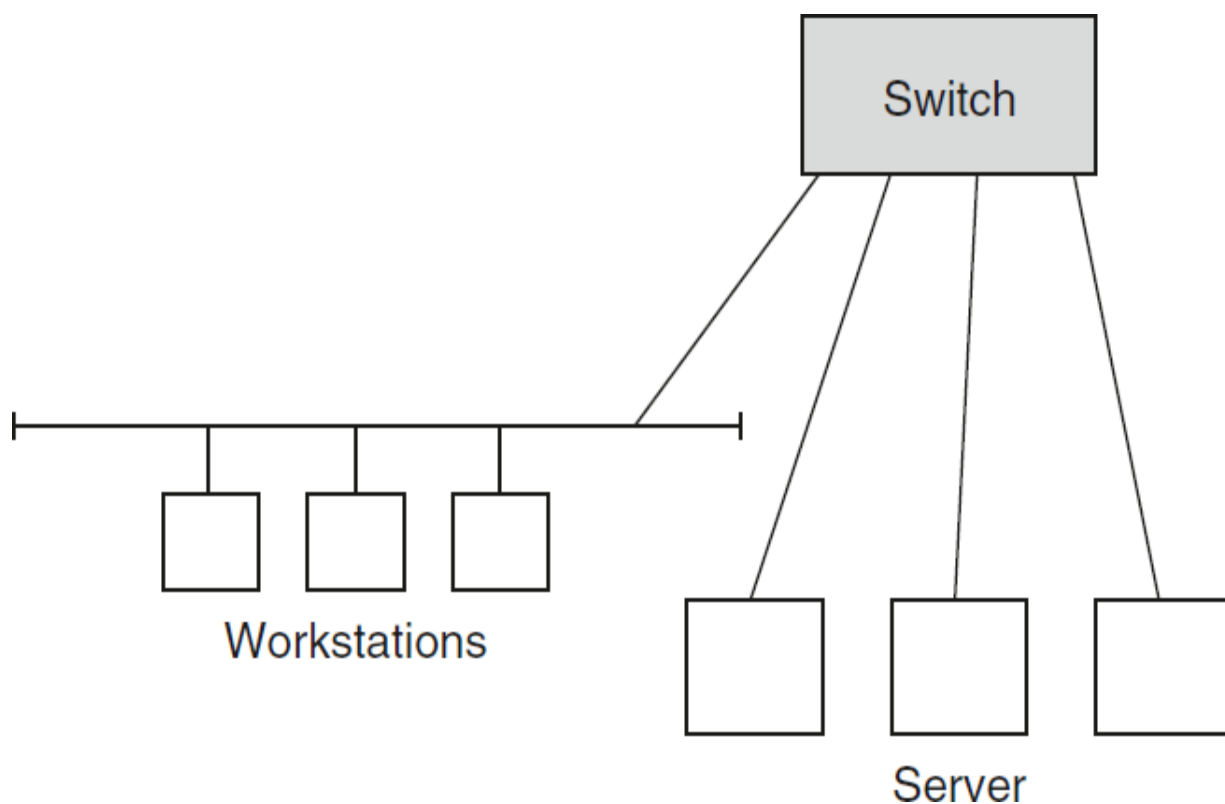


Abb. 1–8 *Switch-Prinzip mit separaten Ports*

Switching-Verfahren

Werden Portgruppen eines Switches direkt einem einzelnen Netzwerksegment mit mehreren Stationen zugeordnet, so spricht man von Port- oder statischem Switching.

Das dynamische Switching hingegen interpretiert zunächst das auf einem bestimmten Port empfangene Datenpaket (*Frame*), bevor anschließend die Weiterleitung an den im Frame-Header angegebenen Zielport (bzw. Zieladresse) erfolgt. Für einen bestimmten Zeitraum existiert also eine Port-zu-Port-Verbindung unter Ausnutzung der insgesamt möglichen Netzwerkbandbreite. Anschließend wird diese Verbindung wieder getrennt und eine neue Port-zu-Port-Verbindung aufgebaut. Es erfolgt demnach ein ständiges Wechseln der Port-zu-Port-Verbindungen.

Beim Cut-Through wird ein ankommendes Datenpaket nicht vollständig gepuffert und überprüft, sondern lediglich das Feld, aus dem die Zieladresse hervorgeht, ausgewertet. Danach wird der Frame ohne Verzögerung *on the fly* unverzüglich durchgereicht. Dies stellt ein äußerst schnelles Verfahren dar. Man muss allerdings bedenken, dass hierbei keine CRC-Prüfung vorgenommen wird. Fehler in der Prüfsumme werden also nicht erkannt.

Beim Store-and-Forward-Verfahren, das von den meisten Switch-Herstellern verwendet wird, erfolgt eine Pufferung des gesamten Frames; dabei kann eine ordnungsgemäße Überprüfung vorgenommen werden. Die Zeitspanne, die ein Frame zwischen Empfang am Eingangsport und Verlassen am Ausgangsport eines Switches benötigt, wird als »Latency Delay« bezeichnet. Dieser Wert wird maßgeblich für die Geschwindigkeitsmessung von Switches verwendet.

Die Switching-Technologie hat sich bereits seit einigen Jahren in den Unternehmensnetzwerken etabliert. Dennoch existiert kein allgemeingültiger Standard, da zum Teil nach wie vor herstellerspezifische Besonderheiten zum Einsatz kommen. Eine Differenzierung einzelner Switching-Typen erfolgt durch den ISO/OSI-Layer, auf dem die Datenübertragung stattfindet.

Die zuvor beschriebenen Switching-Verfahren beziehen sich ausschließlich auf den ISO/OSI-Layer 2, der zur Adressierung von Sender und Empfänger MAC-Adressen (MAC = *Media Access Control*), jedoch keine IP-Adressen verwendet.

In den letzten Jahren hat sich der Trend vom Layer-2-Switching zum Layer-3/4-Switching fortgesetzt. Diese Entwicklung lässt sich größtenteils dadurch erklären, dass der Anteil an *nicht routbaren* Netzwerkprotokollen immer mehr

zurückgeht und stattdessen vermehrt TCP/IP-basierte Protokolle eingesetzt werden. Somit ergibt sich natürlich ein Bedarf nach einer Layer-3-Datenstrukturierung, die ausschließlich durch Routing oder Layer-3/4-Switching zu realisieren ist.

Ein reines Layer-3-Switching gibt es nicht, da zur Erreichung der gewünschten hohen Transportgeschwindigkeit die aufwendige Routing-Funktionalität des jeweils eingesetzten Switches mit seinen schnellen Layer-2-Switching-Techniken kombiniert werden muss. Aufgrund des fehlenden allgemeingültigen Standards haben unterschiedliche Hersteller verschiedene Switching-Techniken eingeführt. Als wichtigste dieser Techniken sind das *Fast IP* (3Com), *IP-Switching* (Ypsilon) und das *Tag Switching* (CISCO) zu nennen.

Fast IP geht grundsätzlich von einer direkten Zuordnung von IP-Adresse und MAC-Adresse aus und basiert hauptsächlich auf dem *Next Hop Resolution Protocol* (NHRP; gemäß RFC 2332 vom April 1998). Durch Einsatz dieses Protokolls ist eine Kommunikation zwischen Stationen in verschiedenen Subnetzen möglich, ohne einen Router zu verwenden. Allerdings müssen sich diese unabhängigen Subnetze auf demselben physikalischen Netzwerksegment befinden.

Wird ein längerer zusammengehörender Datenstrom identifiziert, so wird dieser beim *IP-Switching* als *IP-Conversation* interpretiert und anschließend mit hoher Geschwindigkeit – gemäß Layer-2-Spezifikation – geschwitcht. Daten, die nicht als IP-Conversation klassifiziert werden, unterliegen den normalen Routing-Funktionen.

Durch Verwendung des *Tag Switching* ist eine Priorisierung des Datenstroms möglich. Die durch besondere *Tags* gekennzeichneten Datenpakete (Router nehmen diese Kennzeichnung vor) werden beim Transport über Layer-2-Switches bevorzugt.

Eine zunehmende Entwicklung hin zu hohen Bandbreiten im Netzwerk (Video- und Audio-Streaming, Intranet, Internet, Voice over IP usw.) führt unweigerlich zu Konfliktsituationen mit unternehmenskritischen Anwendungen im gleichen Netzwerk. Die Notwendigkeit, für solche Prozesse ausreichend Netzwerkkapazitäten zur Verfügung stellen zu müssen, scheitert zumeist an der Unfähigkeit, die verschiedenen Datenströme differenzieren zu können. Layer-2- bzw. Layer-3-Switches sind dazu nicht in der Lage.

Kann jedoch die Layer-4-Information eines Datenpakets ausgewertet werden, so ist eine Priorisierung der

verschiedenen Anwendungen möglich. Das Layer-4-Protokoll TCP (aber auch UDP) beinhaltet in seinem Header eine sogenannte *Portnummer*, die eine exakte Zuordnung des Datenpakets zur jeweils verwendeten Applikation zulässt: Die Portnummer 3299 gibt beispielsweise an, dass dieses Datenpaket zu einer SAP-Transaktion gehört, Port 23 weist auf eine TELNET-Sitzung hin. Diese Priorisierungsfunktionalität wird auch als *Quality of Service* bezeichnet.

Switch-Typen

Heute unterscheidet man grundsätzlich drei Switch-Typen hinsichtlich ihrer Administration und ihres Funktionsumfangs:

Unmanaged Switches sind Switches, die in der Regel vorkonfiguriert sind und über einen sehr eingeschränkten Funktionsumfang verfügen. Sie sind zwar ohne Konfigurationsaufwand sofort einsetzbar, stellen somit aber auch nur ein Minimum an Flexibilität dar und sind oft für den Einsatz in Unternehmensnetzwerken nicht oder nur eingeschränkt verwendbar. Sie stellen die preiswerteste Alternative dar.

Smart Switches hingegen verfügen bereits über eine angemessene »Intelligenz« und sind gegenüber den unmanaged »Plug-and-Play«-Modellen konfigurierbar. Sie sind oft über Web-

Interfaces mit einer einfachen Administrationsfunktion ausgestattet und erlauben die Implementierung einiger wichtiger Funktionen für zumeist kleinere Unternehmensnetzwerke. Dies können bereits Parameter für »Quality of Service« (QoS) sein, aber auch die Konfiguration einiger VLANs (siehe nächster Abschnitt).

Managed Switches bilden das Nonplusultra in dieser Reihe von Switch-Typen. Sie sind vollständig administrierbar, besitzen professionelle Schnittstellen für die Konfiguration und sind für den Einsatz in komplexen Netzwerken gedacht. Alle für Unternehmensnetzwerke relevanten Funktionen werden unterstützt, einschließlich umfangreicher Sicherheitsfunktionen.

Virtuelle LANs (VLAN)

Zur besseren Strukturierung »geswitchter« Ressourcen können mit den heute verfügbaren Switches sogenannte *virtuelle LANs* (VLANs) gebildet werden. Einzelne Netzwerkkomponenten (Server, Router, LAN-Segmente usw.) können damit in Gruppen zusammengefasst werden und bilden somit ein virtuelles LAN, das sogar über Switch-Grenzen hinaus definiert werden kann.

So ist es beispielsweise möglich, die Server von Switch A, die den Ports 1, 2, 3 und 7 zugeordnet sind, sowie die Workstations

von Switch B, die an die Ports 5, 6 und 7 herangeführt wurden, zu einem einzigen VLAN zusammenzufassen. Damit bilden diese physikalisch unabhängigen Komponenten eine Art »Zweckgemeinschaft«. Man spricht bei diesen virtuellen LANs auch von *Broadcast Domains*; denn die Broadcasts verbleiben in den virtuellen Grenzen des VLAN. Dabei wird bei der Festlegung und Umsetzung zwischen verschiedenen VLAN-Typen unterschieden:

- *Port-Grouping-VLANs*

Beim Port-Grouping-VLAN werden einzelne Ports eines oder mehrerer Switches zu einem VLAN zusammengefasst (siehe [Abb. 1–9](#)). Die Strukturierung ist zwar recht einfach, verhindert jedoch bei geografischer Veränderung von Netzwerkressourcen (PCs, Router, Workstations werden an anderer Stelle im Netzwerk installiert, ziehen also um) eine dynamische Rekonfiguration des VLAN.

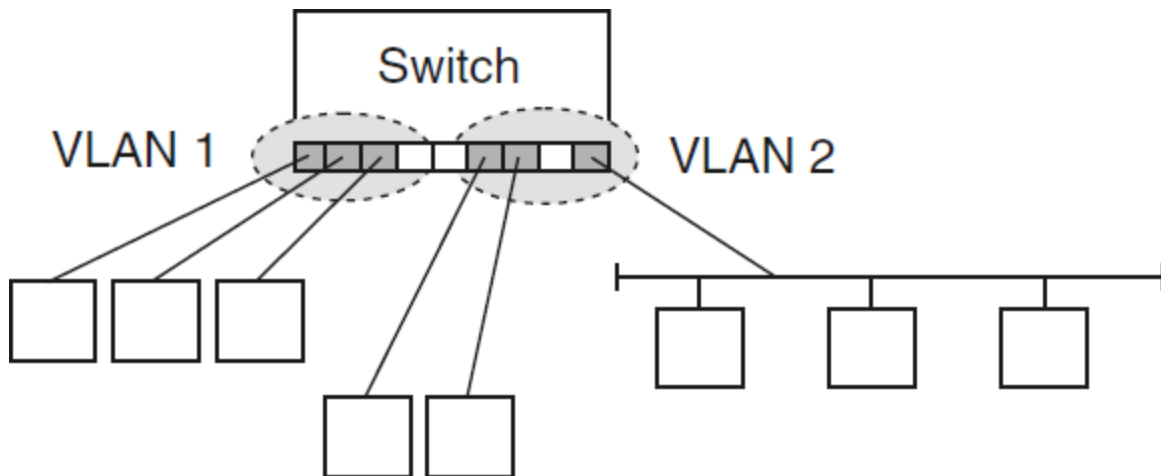


Abb. 1–9 Prinzip eines Port-Grouping-VLAN

- *MAC-Grouping-VLANs*

Das MAC-Grouping-VLAN orientiert sich (wie der Name schon vermuten lässt) an der MAC-Adresse einer Netzwerkkomponente (*Network Interface Controller* = NIC). Ein MAC-Layer-orientiertes VLAN lässt jederzeit eine dynamische Rekonfiguration im Segment zu. Es ist daher völlig unerheblich, ob eine Netzwerkkomponente heute Port 2 des Switches A und morgen Port 8 des Switches B zugeordnet wird. Gruppierungskriterium für das VLAN ist immer die eindeutige MAC-Adresse des jeweiligen Geräts.

- *Network-Grouping-VLANs*

VLANs auf Basis eines Network Grouping werden durch Layer-3-Switches realisiert (der Network Layer wird im ISO/OSI-Referenzmodell als der Layer 3 bezeichnet), in denen sämtliche Netzwerkkomponenten zusammengefasst sind, die das Netzwerkprotokoll IP verwenden. Diese Funktion ist

allerdings nicht mit dem originären *Routing* zu verwechseln, in dem Routing-Protokolle (OSPF, RIP2 usw.) auf dedizierten Routern eingesetzt werden.

Es handelt sich also hierbei eigentlich um eine flache, »gebridgte« Netzwerktopologie, in der zumeist der aus dem Ethernet bekannte Spanning-Tree-Algorithmus verwendet wird. Dabei ist es jedoch erforderlich, dass sich die zu einem Layer-3-VLAN komprimierten Netzwerkkomponenten Protokollen bedienen, die als routbar bezeichnet werden. Dazu gehören beispielsweise die Protokolle der TCP/IP-Protokollfamilie, aber auch DECnet oder IPX. So können Rechner oder Server, die lediglich über das in der Vergangenheit in der Windows-Welt sehr oft eingesetzte NetBIOS-Protokoll verfügen, kein Bestandteil dieses VLAN sein, da NetBIOS kein routbares, sondern lediglich ein »bridgefähiges« Protokoll ist, also auf dem ISO/OSI-Layer 2 transparent von einem in das andere Netzwerksegment übertragen wird.

- *IP-Multicast-Grouping-VLANs*

Eine etwas eigenwillige Art und Weise, VLANs zu definieren, wird durch das IP-Multicast-Grouping dokumentiert. Hier werden einzelne Netzwerkkomponenten einer bestimmten Multicast-Gruppe zugeordnet, deren Kommunikation untereinander auf das somit gebildete VLAN begrenzt wird.

Ein per Multicast verschicktes Datenpaket wird lediglich von den Mitgliedern der definierten Multicast-Gruppe empfangen. Andere IP-Knoten (die nicht zu diesem VLAN gehören) erhalten das Datenpaket nicht.

Gateway

Ein *Gateway* verbindet zwei Netzwerke mit unterschiedlicher Protokollstruktur. Dabei erfolgt in der Regel eine Protokollumsetzung über mehrere Schichten des ISO/OSI-Schichtenmodells. Soll beispielsweise eine Kommunikation zwischen einem prozessnahen, technischen Netzwerk mit zahlreichen unterschiedlichen Sensoren oder Geräten mit einem kommerziellen Netzwerk verbunden werden (beispielsweise zur Auswertung von Sensordaten für einen Produktionsprozess), so ist eine direkte Kommunikation zunächst nicht möglich, da die Komponenten aus beiden Netzwerken »in unterschiedlichen Protokollwelten leben«. Wird allerdings zwischen beiden Netzwerken eine Instanz geschaffen, die auf TCP/IP-Seite Daten entgegennehmen kann und diese anschließend in geeigneter Form aufbereitet, so ist damit ein Gateway geschaffen, das zwischen den beiden Protokollwelten vermittelt.

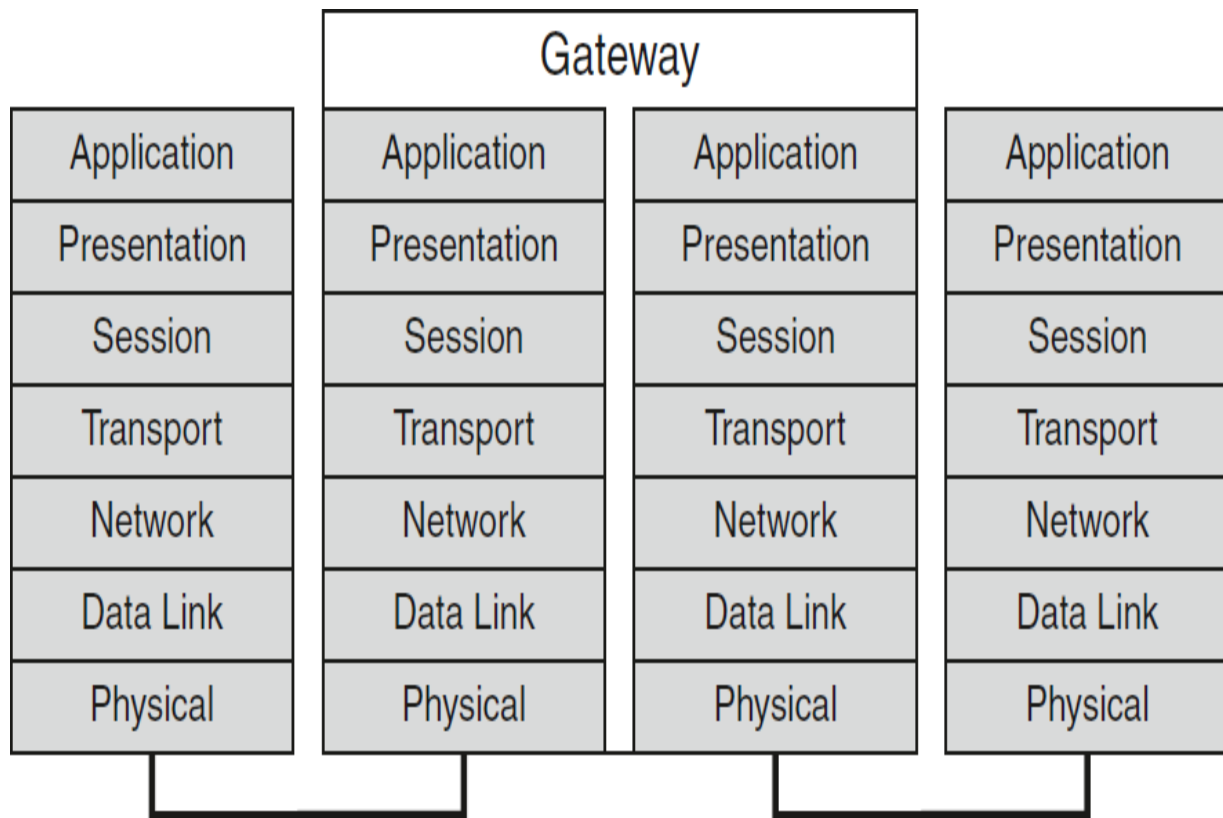


Abb. 1–10 Funktionsaufbau eines Gateways

Router

Router werden immer originär auf dem Layer 3 des OSI-Referenzmodells eingesetzt und übernehmen die Aufgabe, Datenpakete aufgrund ihrer Adressinformationen den Zielrechnern und Anwendungen zuzustellen. Dies geschieht durch ausgeklügelte Verfahren der Wegewahl (Wahl des Weges der Datenpakete), das sogenannte *Routing*. Die einzelnen Datenpakete werden dabei also nicht mehr transparent durchgereicht, sondern explizit nach konkreten Routing-Algorithmen und -Vorschriften dem Empfänger zugestellt.

Router sind im Gegensatz zu Brücken keine logischen Verbindungskomponenten verschiedener LANs zu einem einzigen LAN, sondern stellen das vermittelnde Bindeglied zweier (oder auch mehrerer) logisch unabhängiger Netzwerke dar. In vielen Fällen verbinden sie jedoch auch physikalisch unterschiedliche (heterogene) Netzwerke.

Teilnehmer des jeweils anderen Netzwerks müssen explizit adressiert werden (z.B. mit IP-Adressen), um untereinander Kontakt aufnehmen zu können. Datenpakete, die der Router erhält, werden von ihm gelesen und auf ihre Adressaten hin überprüft. Wenn der Router diese kennt bzw. seine Routing-Informationen ausreichen, so leitet er das Datenpaket sofort weiter; andernfalls kontaktiert er weitere Router, mit denen er ggf. in Verbindung steht. Von diesen erhält er dann die notwendige Routing-Information, und die Datenpakete können zugestellt werden.

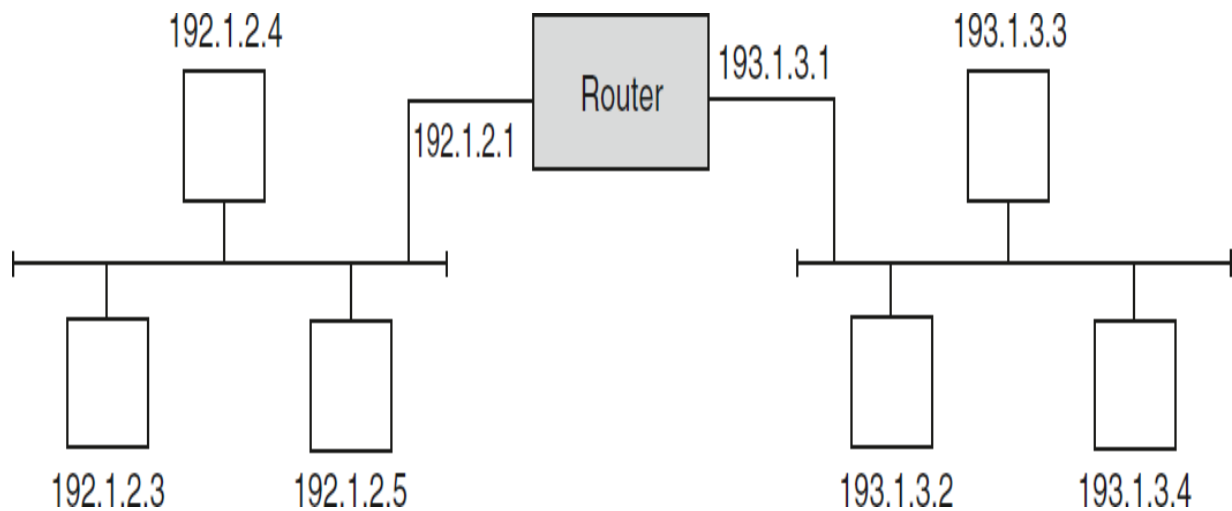


Abb. 1–11 *Prinzip der Wegewahl (Routing) beim Einsatz eines Routers*

HINWEIS

Weitergehende und ausführliche Angaben zum Einsatz von Routern und den damit verbundenen Möglichkeiten enthält [Kapitel 4.](#)

TCP/IP – Grundlagen

Während man sich früher relativ selten dafür entschied, bereits zu Beginn einer Netzwerkplanung Netzwerkprotokolle der TCP/IP-Protokollfamilie einzusetzen, ist es heute längst Normalität, TCP/IP als Basis jeglicher Unternehmenskommunikation anzusehen. Infolge des gezielten Einsatzes bestimmter proprietärer Netzwerkprotokolle in der Vergangenheit (z.B. SNA, DECnet oder AppleTalk) – oft in Abstimmung mit der nicht immer sachbezogenen Unternehmenspolitik – ging die notwendige Beurteilungsfähigkeit über die globale Netzwerkstruktur des Unternehmens oft völlig verloren. In autonomen Bereichen (technischer und kaufmännischer Bereich) entstanden voneinander unabhängige Netzwerke, jedes abgestimmt auf die lokalen Gegebenheiten und besonderen Anforderungen. Erst nach Jahren führten die notwendigen Kommunikationserfordernisse der Bereiche untereinander dazu, dass man sich über eine die Topologie übergreifende Kommunikation Gedanken machte.

Spätestens zu diesem Zeitpunkt sprach man immer öfter von TCP/IP (*Transmission Control Protocol/Internet Protocol*) als »Netzwerkintegrator«. Mit ihm ließen sich zwei äußerst wichtige strategische Ziele realisieren:

- zentrales Netzwerkmanagement auf IP-basierten Management-Tools für heterogene Netze
- optimierte Verfügbarkeit von Anwendungen auf von der Topologie unabhängigen Plattformen

Beide Ziele konnten mithilfe des Netzwerkprotokolls TCP/IP und seiner Protokollfamilie in der Praxis effektiv umgesetzt werden. Dafür sorgte eine Vielzahl an Anwendungen, die fast immer im Lieferumfang der bedeutenden TCP/IP-Implementierungen enthalten sind bzw. waren: TCP/IP-Komponenten der Windows-Betriebssysteme, der UNIX-Plattformen namhafter Hersteller sowie der mittlerweile populären Linux-Distributionen wie Debian, openSUSE oder Ubuntu (um nur einige wenige zu nennen). Leistungsfähige Anwendungen bzw. Anwendungsprotokolle wie TELNET und FTP decken auch heute in der Regel viele Basisanforderungen ab (Remote Login, Konsolenbetrieb, Dateitransfer usw.), wurden aber mittlerweile häufig durch sichere Erweiterungen ersetzt (z.B. Secure Shell statt TELNET oder auch SFTP). Für das Netzwerkmanagement ist das Protokoll SNMP von großer Bedeutung. Damit lassen sich sämtliche im Netzwerk definierten IP-Knoten über ein komplexes Agentensystem verwalten und steuern. Darüber hinaus basieren heutige Kommunikationstechnologien (z.B. Voice over IP, Audio- und Videostreaming) sowie die moderne

Anwendungskommunikation (z.B. SAP) fast ausschließlich auf Protokollen der TCP/IP-Familie.

Wesen eines Protokolls

Bevor die TCP/IP-Protokollfamilie nachfolgend in allen Einzelheiten erläutert wird, soll an dieser Stelle zunächst einmal ganz allgemein auf die wesentlichen Funktionen eines Protokolls eingegangen werden:

Stellen Sie sich ein Telefongespräch vor. In der Regel sind zwei Personen beteiligt: der Anrufer und der Angerufene (Sender und Empfänger). Die intuitiv ablaufenden Regularien eines Gesprächs werden mittlerweile nicht mehr bewusst wahrgenommen. Der Anrufer beginnt die Kommunikation, er ist Initiator. Er spricht den Angerufenen an, nachdem er ihm seinen Namen mitgeteilt hat (Identifikation). Der Angerufene reagiert entsprechend. Entweder kennt er den Anrufer, dann begrüßt er ihn sofort, oder aber er kennt ihn nicht. In letzterem Fall ist eine mehr oder weniger umfangreiche Phase des Kennenlernens erforderlich, um das Gespräch fortzusetzen.

Ein Protokoll oder ein Netzwerkprotokoll besitzt den gleichen konzeptionellen Ansatz. Ein Rechner A will mit einem Rechner B kommunizieren. Sind beide Rechner einander

bekannt, so können sich diese »unterhalten«. Kennt aber einer von beiden seinen Kommunikationspartner nicht, so kommt kein Gespräch zustande. Das Kennenlernen erfolgt z.B. über eine *Routing-Tabelle*. In ihr sind sämtliche im Netzwerk relevanten Wege (Routen) eingetragen. Befinden sich zwei kommunizierende Rechner im selben Netz, so kann direkt miteinander Kontakt aufgenommen werden. Netzübergreifende Kommunikation kann jedoch nur über ihre Vermittlungsstationen, die Router (oder auch Gateways), erfolgen. Eine so definierte Wegewahl stellt einen Routing-Eintrag dar. Kommt eine Kommunikationssitzung (bzw. ein Telefongespräch) zustande, so können Daten übertragen werden (bzw. das Gespräch kann stattfinden). Die Beendigung der Sitzung wird von einem der beteiligten Partner eingeleitet, der andere Partner bestätigt und die Session wird geschlossen.

Das zuvor geschilderte Telefongespräch bzw. die Sitzung (*Session*) spielte sich auf einem hohen Protokolllevel ab: Beide Gesprächspartner brauchten sich nicht darum zu kümmern, wie die physikalischen Gegebenheiten (also die darunter liegenden Level) ausgestaltet sind, um das Gespräch zu realisieren. Die Telefone und die Telefonleitung standen als vorausgesetzte Hilfsmittel zur Verfügung.

Betrachtet man die Gesetzmäßigkeiten einer solchen Kommunikationsform etwas genauer, so können folgende Charakteristika genannt werden:

- Jede übermittelte Information muss vom Gesprächspartner bestätigt werden. Es ist nicht damit getan, die Informationen einfach zu übermitteln. Verständnisprobleme oder auch Störungen bei der Übertragung könnten den Informationsgehalt verändern und somit unbrauchbar machen. Diese Form der Kommunikation kann daher als besonders sicher gelten.
- Die Fülle an Informationen (zahlreiche Bestätigungen), die während eines Telefongesprächs übermittelt wird, bedingt eine sehr gute Leitungskapazität. Andernfalls treten auch hier ggf. Störungen auf, und die zu übertragenden Informationen erreichen den Empfänger nur unvollständig (hohe Bandbreite).
- Beide Kommunikationspartner sind zu jedem Zeitpunkt über den aktuellen Status der Kommunikation im Bilde. Statusinformationen brauchen nicht auf irgendeinem Wege angefordert zu werden (gute Orientierung innerhalb der Kommunikation).
- Ein Telefongespräch vollzieht sich in drei Phasen: Zunächst wird der Gesprächspartner angewählt (Verbindungsaufbau), dann beginnt das eigentliche Gespräch (Datenübertragung).

Schließlich beendet ein Gesprächspartner die Kommunikation durch Auflegen (Verbindungsabbau).

Protokolle, die eine Kommunikation in der beschriebenen Art und Weise steuern, werden *verbindungsorientierte Protokolle (connection oriented)* genannt. Ein klassisches Beispiel für ein verbindungsorientiertes Protokoll ist TCP.

Analog zu den verbindungsorientierten Protokollen gibt es *verbindungslose Protokolle (connectionless oriented)*. Sie weisen im Einzelnen folgende Merkmale auf:

- Es existiert ein einfacher Datenverkehr (Datagrammverkehr) zwischen den Kommunikationspartnern. Die einzelnen Datenpakete sind über Quell- und Zieladresse identifizierbar und können so zugestellt werden.
- Da keine speziellen Sicherungsmechanismen existieren, kann diese Kommunikationsform als äußerst schnell bezeichnet werden.
- Fehlende Sicherheit bedeutet jedoch auch eine erhöhte Störanfälligkeit:
 - Datenpakete können verloren gehen.
 - Datenpakete werden in falscher Reihenfolge zum Ziel transportiert.
 - Datenpakete können doppelt empfangen werden.

- Um überhaupt einen einigermaßen verlässlichen Informationsfluss bei verbindungslosen Protokollen zu gewährleisten, müssen übergeordnete Protokolle eine ausreichende Sicherung durchführen (wie beispielsweise TCP).

Verbindungslose Protokolle werden daher häufig für die Übertragung von Statusinformationen im Netzwerk verwendet. So benutzt beispielsweise SNMP (*Simple Network Management Protocol*) das Protokoll UDP (*User Datagram Protocol*), um Managementinformationen zwischen einzelnen Netzwerkkomponenten und Managementsystemen zu übermitteln.

In Analogie zum Telefongespräch als verbindungsorientierte Übertragung kann für ein verbindungsloses Protokoll beispielhaft der Versand und Empfang von Briefen genannt werden. Nach Einwurf eines Briefes in den Postkasten des Übermittlers wird dieser von ihm abgeholt und nach einem festgelegten Transportverfahren zum Briefkasten des Empfängers gebracht. Hier kann ihn der Empfänger entgegennehmen. Während des Transports existiert keinerlei Verbindung der beiden Kommunikationspartner untereinander. Die Ankunft des Briefes ist demnach ungewiss; auch wenn man eine gewisse Verlässlichkeit des Transporteurs stillschweigend

voraussetzt, so bietet dieser keinerlei Garantien, da der Brief auf dem Transportweg auch durchaus verloren gehen kann.

Eine genaue Definition des Begriffs *Protokoll* ist im EDV-technischen Sinne nur schwer möglich. Das Protokoll wird hier, völlig anders als im normalen Sprachgebrauch, als eine Zusammenstellung verschiedener Regeln und Vorschriften verstanden, um eine bestimmten Anforderungen genügende Kommunikation zwischen zwei oder mehreren Partnern zu ermöglichen. Die Kommunikationspartner können dabei Rechner, Netzwerkkomponenten (z.B. Router oder Switches) oder auch nur einfache Anwendungsprogramme sein, die im Netzwerk verteilt miteinander kommunizieren.

Low-Layer-Protokolle

Es gibt für die unterschiedlichsten Bedarfssituationen und Leistungsspektren eigens entwickelte Protokolle. Sie können allerdings nur dann die gewünschten Ergebnisse liefern, wenn sie aufeinander abgestimmt sind. Die wichtigsten Protokolle und ihre Eigenschaften sind im Folgenden dargestellt.

Protokolle der Datensicherungsschicht (Layer 2)

Auf der Ebene der Datensicherungsschicht (Layer 2) werden die Protokolleigenschaften des MAC-Teil-Layers (*Media Access*

Control) und des LLC-Teil-Layers (*Logical Link Control*) genau spezifiziert. So wird innerhalb des LLC ein Zuordnungsverfahren beschrieben, das die Verwendung mehrerer Protokolle (Multi-Protokollstack) an einer Station im Netz ermöglicht: der *Service Access Point* (SAP). Das Zuordnungsverfahren für herstellerspezifische, nicht genormte Protokolle (wie z.B. IP) wird über ein *SNAP* (*Subnetwork Access Protocol*) abgewickelt (siehe [Abb. 2-1](#)).

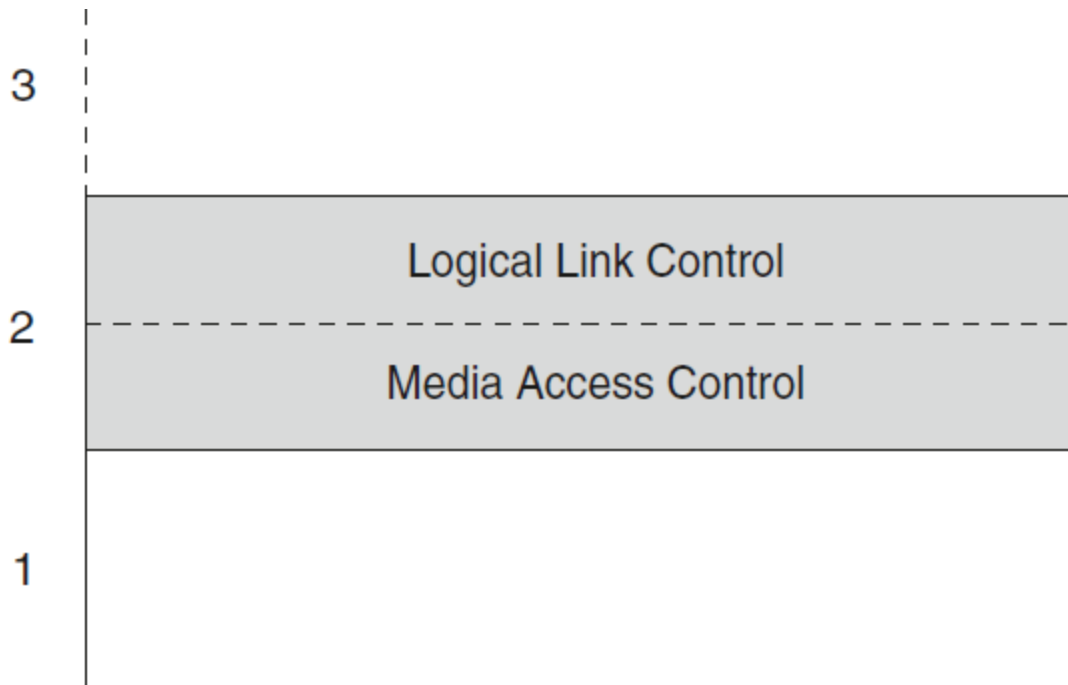


Abb. 2-1 *Layer-2-Schicht (MAC und LLC)*

Media Access Control (MAC)

Media Access Control übernimmt (bei Ethernet ebenso wie bei Token Ring) die Zugriffssteuerung für das jeweils verwendete

Medium. Sie ist für die Adressierung der einzelnen Netzwerkstationen zuständig und überwacht ihren Zustand. Grundsätzlich wird zwischen drei verschiedenen Typen oder Formen von MAC-Adressen unterschieden:

- *Adressen zur eindeutigen Identifizierung*

Herstellerspezifisch werden weltweit eindeutige *burnt-in addresses* vergeben und in den Netzwerkcontrollern auf Chip-Bausteinen hinterlegt. Sie haben sechs Bytes Länge, von denen die ersten drei Bytes für herstellerspezifische Informationen reserviert sind. Die restlichen Bytes werden für eine fortlaufende Serien- bzw. Fabrikationsnummerierung verwendet. Die Hardwareadressen der Token-Ring-Controller, die auch *universal address* genannt werden, lassen sich durch *local addresses*, also selbst definierte MAC-Adressen, überschreiben und ermöglichen somit den Aufbau einer eigenen Adress-Struktur. Per Definition ist für das erste Byte einer *local address* ein Bereich zwischen den Hex-Werten 40 und 7F vorgesehen. In den ersten drei Bits des ersten Bytes wird u.a. das RII-Bit (*Routing Information Indicator*) oder auch *Source-Routing-Bit* und das U/L-Bit (*Universal/Local Address Bit*) untergebracht. Der Rest (bestehend aus 5 weiteren Bytes) kann bei einer *local address* durch eigene beliebige Hex-Codes ergänzt werden.

- *Broadcast-Adressen*

Eine Broadcast-Adresse besitzt lediglich nur den Hex-Wert FFFFFFFF. Diese Adresse ist gewissermaßen als Zieladresse zu verstehen, denn sie wird von allen Controllern der Stationen im Netzwerk empfangen und normalerweise bis zur Netzwerksoftware durchgereicht. Dieser Vorgang soll dem Sender die Möglichkeit geben, jede Netzwerkstation erreichen zu können, um von ihr Informationen zu erhalten bzw. ihr Informationen zustellen zu können.

- *Multicast-Adressen*

Über diesen speziellen Adresstyp wird in der Regel versucht, zwischen Netzwerkkomponenten bestimmter Kategorie und mit speziellen Eigenschaften (auch bestimmter Hersteller) nur für sie bestimmte Informationen auszutauschen. Die Kommunikation zwischen Routern wird beispielsweise über Multicast-Adressen realisiert. Die Netzwerklast wird dadurch – unter Vermeidung von Broadcasts – erheblich vermindert.

Logical Link Control (LLC)

Der LLC-Layer ist, im Gegensatz zum MAC-Layer, vollkommen unabhängig vom physikalischen Medium. Ob es sich um ein CSMA/CD-Netzwerk (IEEE 802.3) oder um ein Token-Ring-Netzwerk (IEEE 802.5) handelt, spielt keine Rolle. Er stellt einen

eigenen Standard in der IEEE-802.2-Norm dar und ist in Token-Ring- und Ethernet-Frames identisch.

Man unterscheidet in dieser Schicht drei verschiedene LLC-Typen:

- LLC1 Not Acknowledged Connectionless Service (*User Datagram Service*)
- LLC2 Acknowledged Connectionless Service
- LLC3 Connection Oriented Service

Insbesondere die Flexibilität des LLC1-Typs (Punkt-zu-Punkt-, Punkt-zu-Mehrpunkt-Adressierung und Broadcasting) macht ihn für TCP/IP interessant. Auf dieser Protokollschicht ist zwar infolge des Datagrammkonzepts keine Übertragungssicherheit gewährleistet, dies ist jedoch ohnehin Aufgabe des höheren Protokolls.

Der LLC2-Typ ist zwar grundsätzlich mit dem LLC1-Typ identisch, allerdings vollzieht er bereits auf dieser niedrigen Protokollschicht ein umfassendes Acknowledgement. Er ist daher in den meisten Fällen zu aufwendig und kostet viel Zeit.

Der LLC3-Typ arbeitet verbindungsorientiert. Kommunikationspartner etablieren untereinander logische Links, und eine bereits hier einsetzende Steuerlogik erkennt

Fehler und kann diese beheben. Die bekannte Protokollarchitektur aus dem Hause IBM (SNA) bevorzugt diesen LLC-Typ. Protokolle wie SDLC (*Synchronous Data Link Control*) oder HDLC (*High Level Data Link Control*) beruhen auf LLC3.

Die LPDU (*LLC Protocol Data Unit*) besteht aus den beiden SAP-Feldern DSAP (*Destination Service Access Point*) und SSAP (*Source Service Access Point*), dem *Control Field* und dem *Information Field* einschließlich Daten (siehe [Abb. 2-2](#)). Die SAPs stellen Protokollnummern dar, die der ISO-Norm entsprechen und auf das in diesem Frame verwendete Protokoll weisen (die SAPs sind i.d.R. gleich, da die Kommunikation zweier Partner über zwei unterschiedliche Protokolle nicht möglich ist). Handelt es sich jedoch um ein nicht standardisiertes oder herstellerspezifisches Protokoll, so erfolgt der Zugriff auf die Dienste dieses Protokolls über ein sog. SNAP (*SubNetwork Access Protocol*). In dem Fall lauten die Werte im DSAP und SSAP AA.

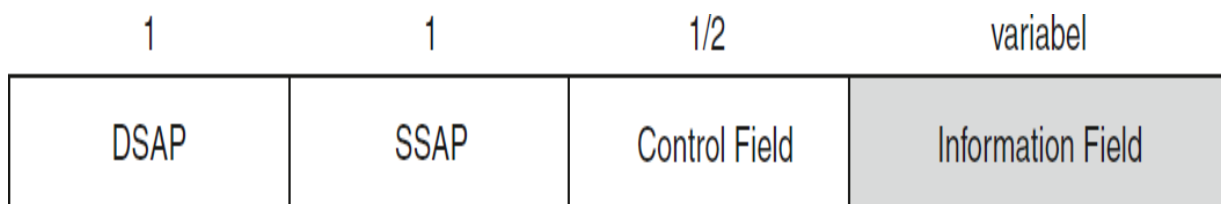


Abb. 2-2 *LLC + Information Field = LPDU*

Für die verbindungslosen (LLC1) und verbindungsorientierten (LLC2, LLC3) LLC-Typen gibt es jeweils unterschiedliche LPDU-Formate. Für die Identifizierung dieser Formate sind Bit 0 bis 2 sowie Bit 4 und 5 des Kontrollfelds (1 Byte) verantwortlich. Bit 3 wird P/F-Bit (Poll-Bit in Command LPDUs, Final-Bit in Response LPDUs) genannt. Im LLC1 (siehe [Abb. 2-3](#)) lassen sich lediglich drei Formate unterscheiden:

- *XID-Command/Responses* (Exchange Identifier) – 101 11
Die sendende Station übermittelt ihrem Empfänger Identifikation und Charakteristik. Sie erwartet eine XID-Response.
- *TEST-Command/Response* – 111 00
Die sendende Station versucht durch Übertragung eines TEST-Commands ihren Empfänger zur Übermittlung einer TEST-Response zu veranlassen. Dieses Verfahren wird zur Ermittlung der Route verwendet.
- *UI-Command* (Unnumbered Information Command) – 000 00
Dieses LPDU-Format ist für den (ungesicherten) Datentransport verantwortlich.

0	1	2	3	4	5	6	7
M	M	M	P/F	M	M	1	1

Abb. 2-3 *Control Field – LLC1*

Im *LLC3* existieren folgende Formate:

- *Information Transfer Format*

Das Kontrollfeld umfasst hier zwei Bytes. Sie nehmen den Sende- und Empfangszähler auf. Der I-Format-Frame ist für die Übertragung von Informationen und Daten zuständig.

- *Supervisory Format (S-Format)*

Das Kontrollfeld umfasst hier ebenfalls zwei Bytes. Das Supervisory-Format dient der Übertragung von Steuerinformationen. Diese sind:

- *Receiver Ready (RR)* – 00

Bereitschaft zum Empfang von I-Format-LPDU

- *Receiver Not Ready (RNR)* – 01

I-Format-LPDUs können zurzeit nicht empfangen werden.

- *Reject (REJ)* – 10

Abweisen von S-Format-LPDUs, damit I-Format-LPDUs erneut empfangen werden können (Retransmission)

- *Unnumbered Format*

Wird für die Übermittlung von zusätzlichen Steuerinformationen verwendet. Dabei werden jedoch Quittungen jeder Art weggelassen. Das 1-Byte-Kontrollfeld (wie im *LLC1*) nimmt verschiedene Commands bzw. Responses auf:

- *Disconnected Mode (DM) Response* – 000 11

Information der Partner-Station, dass zurzeit keine Verbindung aktiv ist

- *Disconnect* (DISC) Command – 010 00

Information der Partner-Station, dass eine Verbindung abgebaut werden soll. Die Partner-Station antwortet mit einem UA bzw. einem DM.

- *Unnumbered Acknowledgment* (UA) Response – 011 00

Antwort auf ein SABME oder DISC (Verbindungsaufbau/-abbau)

- *Set Asynchronous Balanced Mode Extended* (SABME)

Command – 011 11

Station, die einen Verbindungsaufbau wünscht, initiiert einen SABME-Command.

- *Frame Reject* (FRMR) Response – 100 01

Bei Erkennen eines ungültigen LPDU-Formats wird ein FRMR übertragen.

- *Exchange Identification* (XID) Command – 101 11

(siehe LLC1)

- *Test* (TEST) Command/Response – 111 00

(siehe LLC1)

Service Access Point (SAP)

Die Bereitstellung der jeweiligen Protokolldienste innerhalb der LLC-Layer erfolgt über genormte *Service Access Points* (SAP).

Eine Liste der wichtigsten SAPs folgt:

BPDU	42	Bridge Protocol Data Unit (Spanning Tree)
Banyan	BC	Banyan Vines
IBM NM	F4	IBM Network Management
IP	06	Internet Protocol
ISO	FE	International Organization for Standardization
NetBIOS	F0	Network Basic I/O System
Novell	E0	Novell (NetWare)
RPL	F8	Remote Program Load
SNA	04, 05, 08, 0C	Systems Network Architecture
SNAP	AA	SubNetwork Access Protocol
Global	FF	Broadcast
Null	00	IBM SAP Negotiation

Sie werden in den jeweils ein Byte umfassenden Feldern DSAP und SSAP untergebracht (siehe [Abb. 2–4](#)).

0	1	2	3	4	5	6	7
D	D	D	D	D	D	U	I/G

DSAP

0	1	2	3	4	5	6	7
S	S	S	S	S	S	U	C/R

SSAP

Abb. 2–4 *Aufbau des DSAP und SSAP*

Die SAP-Adress-Bits umfassen Bit 0–5. Bit 6 stellt das *User-Defined-Address-Bit* dar, aus dem die Herkunft des SAP hervorgeht. Der Wert »0« bedeutet, dass es sich um einen *User-SAP* handelt, der Wert »1« bezieht sich auf einen SAP nach IEEE. Findet sich im Bit 7 des DSAP der Wert »0«, so liegt eine Individual Address vor, andernfalls eine Group Address. Im SSAP bezeichnet das Bit an dieser Stelle einen Command (Wert »0«) oder eine Response (Wert »1«).

Subnetwork Access Protocol (SNAP)

Die SNAP-Option – als Erweiterung zum LLC – schafft die Möglichkeit, nicht nur Protokolle im SAP der IEEE/ISO-Standards, sondern auch eine Vielzahl herstellerspezifischer oder nicht standardisierter Protokolle abzubilden. Durch dieses

Verfahren wird die ohnehin schon durch SAPs eingeführte Multiprotokollfähigkeit einzelner Netzwerkkstationen erheblich erweitert. Neu entwickelte Protokolle können, bevor sie zum Standard avancieren, bereits unter Verwendung von SNAP in bestehenden Netzwerkstrukturen eingesetzt werden.

Das SNAP-Frame-Format umfasst insgesamt fünf Bytes; die ersten drei Bytes stellen die Kennung des jeweiligen Herstellers bzw. der Organisation, die letzten beiden Bytes die Protokollidentifikation dar. Dieses abschließende 2-Byte-Typenfeld ist mit dem Typenfeld im Ethernet-Frame identisch (siehe [Abb. 2–6](#)).

IEEE 802.3 Frame Format

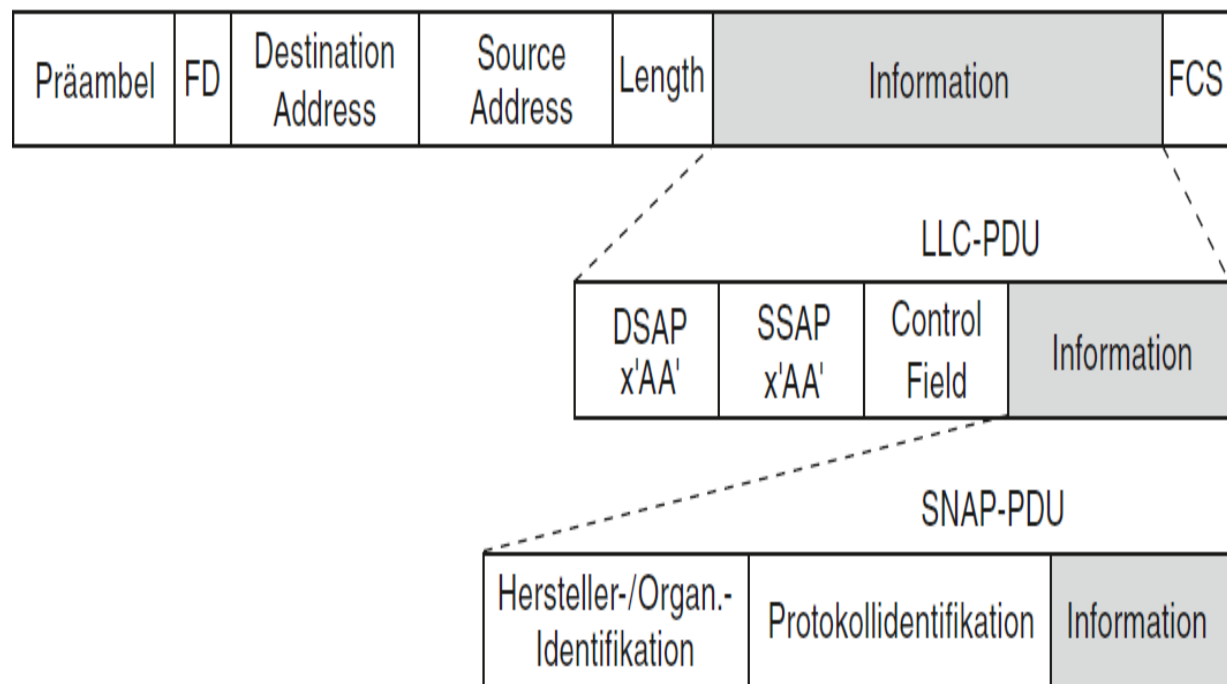


Abb. 2–5 Verschachtelungsprinzip des SNAP-LLC-MAC-Frame

HINWEIS

Lediglich bei vorhandenen *AA-Werten* im DSAP und SSAP existiert ein SNAP-Format. Andernfalls beginnt im Anschluss an die LLC-PDU das Information Field inkl. der Daten.

Protokolle der Netzwerkschicht (Layer 3)

Das ISO/OSI-Schichtenmodell legt eindeutig fest, dass Objekte aus den einzelnen Schichten wegen ihrer Abhängigkeit zueinander im Gesamtkontext gesehen werden müssen. Aus diesem Grund kommt den Protokollen der Netzwerkschicht (Layer 3), zu denen auch das *Internet Protocol* (IP) aus der TCP/IP-Protokollfamilie gehört, eine besondere Bedeutung zu.

Voraussetzungen für die Adressierung von Knoten oder Geräten innerhalb eines Netzwerkverbunds wurden ausführlich behandelt. Nun geht es darum, einige wichtige Protokolle der Netzwerkschicht zu beschreiben und ihre Funktionalität darzustellen. Hierzu gehört das *Internet Protocol* (IP), das *Internet Control Message Protocol* (ICMP), das *Address Resolution Protocol* (ARP), das *Reverse Address Resolution Protocol* (RARP) und Routing-Protokolle, denen allesamt besondere Eigenschaften zukommen.

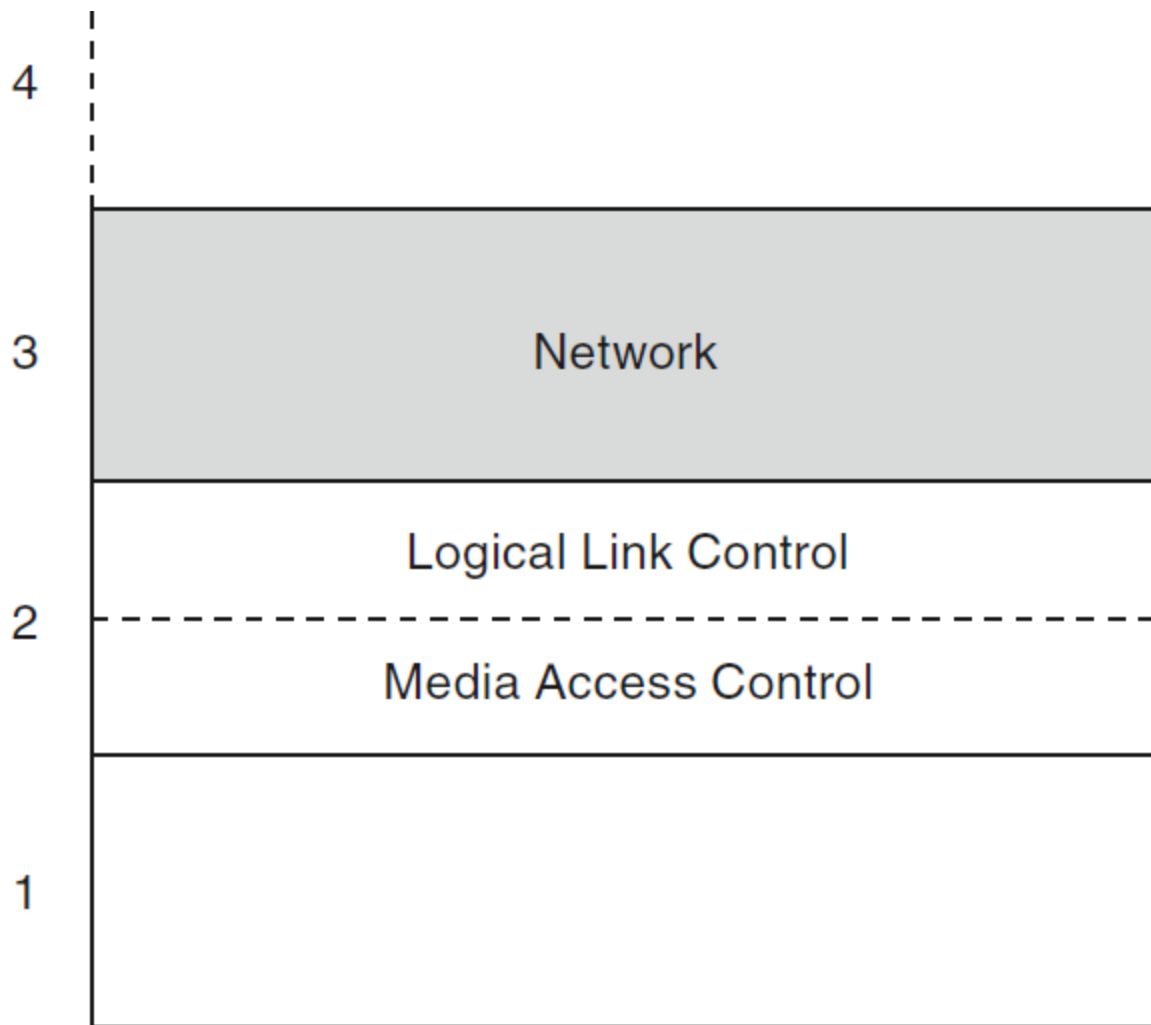


Abb. 2–6 *Layer-3-Schicht (Network)*

Internet Protocol (IP)

Gemeinsam mit dem in Layer 4 angesiedelten *Transmission Control Protocol* (TCP) bildet das *Internet Protocol* als verbindungsloser Datagrammdienst das zentrale Protokollpaar innerhalb der TCP/IP-Architektur. Auf der Ebene des IP-Protokolls werden keine Aufgaben zur Datensicherung

übernommen, dies geschieht auf der übergeordneten Protokollschicht (TCP).

Die Leistungsfähigkeit eines Datagrammverkehrs ergibt sich aus der flexiblen Adressierbarkeit (es können eine, mehrere oder auch alle Stationen im Netz angesprochen werden), der hohen Übertragungsgeschwindigkeit (Datagramme brauchen nicht bestätigt zu werden, Algorithmen zur Fehlerkorrektur fehlen) und einem variablen Routing (die Wegewahl zwischen zwei kommunizierenden Stationen ist nicht festgelegt, sondern kann variieren; Leitungsstörungen können durch Alternativrouten umgangen werden).

Folgende Aufgaben bzw. Merkmale des IP werden im Folgenden ausführlich beschrieben (siehe [Abb. 2-7](#)):

- IP im TCP/IP-Protokollstack
- IP-Datagramme, Aufbau und Funktion
- Fragmentierung
- Maximum Transfer Unit (MTU)

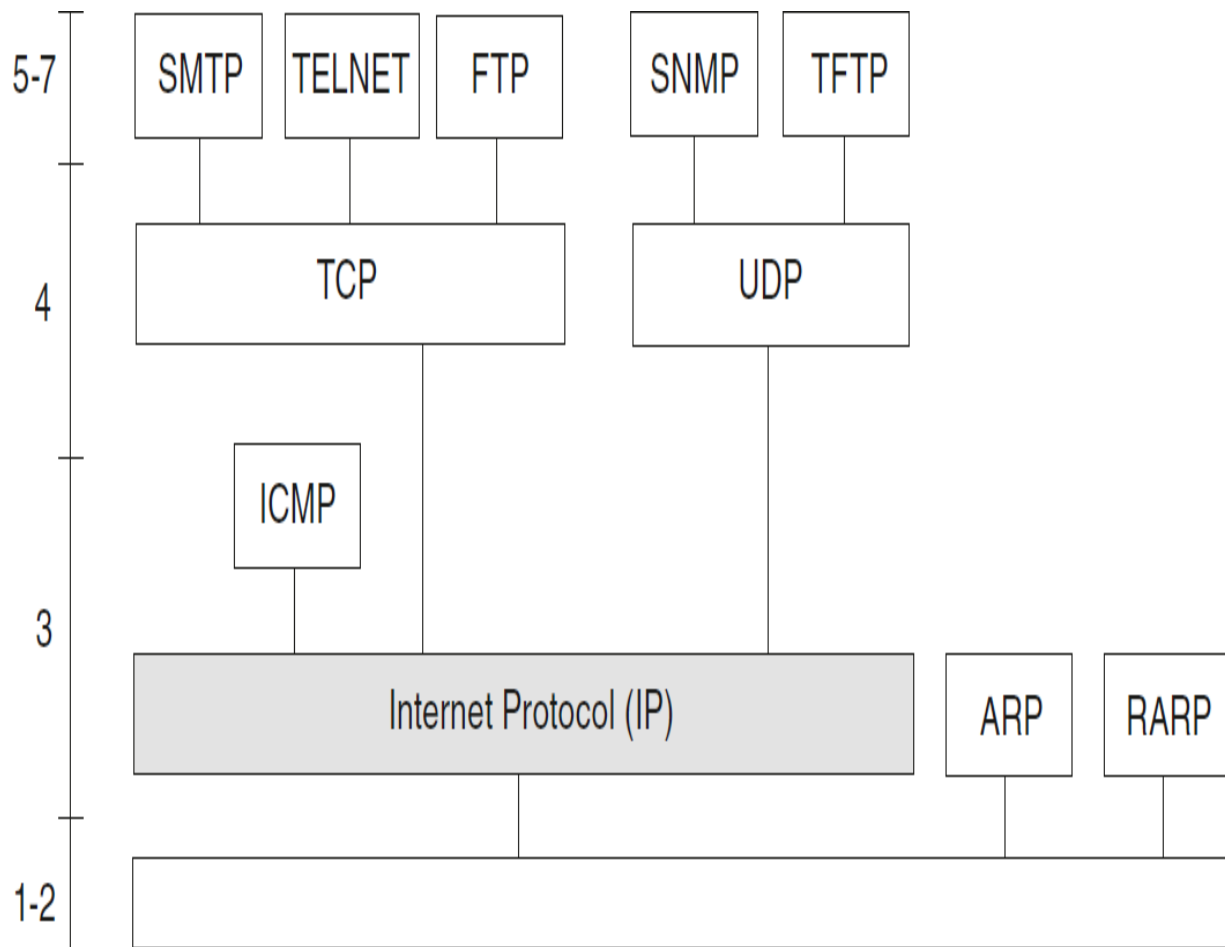


Abb. 2-7 IP im TCP/IP-Protokollstack

HINWEIS

Den Besonderheiten zur IP-Adressierung und der Bildung von IP-Subnetzen wird ein eigenes Kapitel gewidmet ([Kap. 3](#)).

Die Transporteinheit im IP ist das *Datagramm* ([Abb. 2-9](#)). Sein Aufbau wird normalerweise in jeweils 32-Bit-Blöcken vertikal skizziert, wobei der Header mindestens 20 Bytes umfasst (das anschließende Feld *Options* ist variabel und kann bis zu 40

Bytes lang sein) und der restliche Datenteil auf eine Gesamtlänge von bis zu 65.535 Bytes ergänzt wird. In diesem Datenteil wird u.a. auch der TCP-Daten-Frame untergebracht, der aus dem IP-Datagramm schließlich ein TCP/IP-Datagramm werden lässt.

Version	IHL	Type of Service	Packet Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
Options/Padding				
Data				
...				

Abb. 2–8 *Aufbau des IP-Datagramms*

Die einzelnen Felder des IP-Headers (siehe [Abb. 2–8](#)) haben folgende Bedeutung bzw. Funktion:

- *Version*

Festlegung der IP-Versionsnummer eingetragen; Version 4 ist aktuell bzw. wird noch in den meisten Fällen verwendet. Die bereits entwickelte Version 6 (früher IPNG; *IP Next Generation*), wartet u.a. mit einem deutlich erweiterten Adressbereich auf. Datagramme, die von Rechnern verschiedener IP-Versionen erzeugt wurden, lassen sich nicht in den Datenfluss integrieren.

HINWEIS

Nähere Angaben zu IPv6 enthält [Kapitel 3](#).

- *IHL (Internet Header Length)*

Länge des Headers einschließlich der *Options* von 32-Bit-Oktetts (also 4 Bytes). Der Wert 5 würde danach bedeuten, dass lediglich der 20-Byte-Fix-Header existiert, denn dies entspräche $5 \times 32 \text{ Bits} = 160 \text{ Bits} = 20 \text{ Byte}$. Der Maximalwert von Binär 1111, also dezimal 15, weist auf eine Anzahl von $15 \times 32 \text{ Bits} = 480 \text{ Bit} = 60 \text{ Bytes}$ hin.

- *Type of Service*

In diesem 8-Bit-Feld erfolgt eine Beschreibung der Qualität des angeforderten Service nach folgender Aufschlüsselung, wobei die ersten drei Bits den Prioritäten-Level angeben:

000normal

001priority

010immediate

011flash

100flash override

101critical

110internet control

111network control

- Beträgt der Wert der folgenden vier Bits »1«, so sind die Eigenschaften dieser Bits aktiviert. Andernfalls steht die entsprechende Funktion nicht zur Verfügung:

D- Delay (fordert eine Verbindung mit kurzer Verzögerung
Bit an)

T-BitThroughput (fordert hohen Datendurchsatz)

R-BitReliability (fordert hohe Sicherheit, »Discards« sind
seltener)

C-BitCost (fordert Route zu niedrigeren Kosten)

Das letzte Bit bleibt unbenutzt.

- *Total Length*

Gibt die Gesamtlänge des Datagramms an; die Differenz zwischen dem Wert dieses Feldes und dem IHL-Wert ergibt

die Länge des Datenteils im Datagramm. Infolge der Feldlänge von 16 Bits kann hier ein Wert von maximal 65.535 (dezimal) eingetragen werden. Diese Datagrammlänge wird von den meisten Netzwerken nicht unterstützt; somit muss fragmentiert werden. Die generierten Einzelfragmente des Originaldatagramms führen in diesem Feld allerdings nicht mehr den ursprünglichen Wert, sondern beschreiben die Gesamtlänge des vorliegenden Einzelfragments.

- *Identification*

Integer-Wert (16 Bits), der zur Nummerierung fragmentierter Datagramme verwendet wird

- *Flags*

Steuerung der Fragmentierung, sofern sie angewendet wird (3 Bits). Das erste Bit (*high-order bit*) wird nicht benutzt und erhält den Wert »0«. Das mittlere Bit wird auch DF-Bit oder *Do-not-fragment-Bit* genannt. Hier bedeutet der Wert »1«: Ein Fragmentieren ist nicht erlaubt, der Wert »0« bewirkt das Gegenteil. Das dritte Bit (*low-order bit*), auch MF-Bit bzw. »More-Flag-Bit«, gibt an, ob noch ein weiteres Fragment folgt (Wert »1«) oder das letzte Fragment des Datagramms vorliegt (Wert »0«).

- *Fragment Offset*

Zur korrekten Zusammenführung fragmentierter Datagramme ist ein Hinweis erforderlich, der Angaben zur

Position der Daten des jeweiligen Datagrammfragments innerhalb des Originaldatagramms (vor der Fragmentierung) macht. Daher erfolgt hier die Angabe der Anzahl aller vorherigen Datagrammfragmente, die jeweils 8 Oktette (8 Bytes) Daten umfassen. Da sich mit 13 Bits maximal der Wert 8.192 darstellen lässt, ergibt sich daraus die maximale Gesamtlänge von 65.535 Bytes. Das erste Fragment einer Datagrammsequenz erhält an dieser Stelle den Wert »0«.

- *Time*

Korrekt lautet der Name dieses Feldes (8 Bits) *Time To Live* (TTL), mit dem von der sendenden Station die Lebensdauer eines Datagramms vorgegeben wird. Bei jeder Passage durch einen Router wird der Wert reduziert. Sollte während des Datenflusses eines Datagramms durch das Netzwerk sein TTL-Wert »0« erreichen, so wird es beim nächsten Router nicht mehr berücksichtigt (*discarded*). Diese Vorgehensweise verhindert ein endloses Kreisen von Datagrammen im Netzwerk. Ein allgemein verwendeter TTL-Startwert liegt bei etwa 30 Sekunden.

- *Protocol*

Dieses Feld (8 Bits) enthält einen Wert zur Identifizierung des übergeordneten Transportprotokolls (Layer 4), dem der IP-Layer das Datagramm übergeben soll. Folgende Werte sind üblich:

17UDP

6 TCP

1 ICMP

8 EGP

89OSPF

- *Header Checksum*

Sobald ein Router passiert wird, müssen die Werte für TTL, FLAG und FLAG OFFSET geändert werden, sodass sich auch die Checksumme ändert bzw. angepasst werden muss. Zur Vermeidung daraus resultierender, unnötig langer Transportzeiten für Datagramme beschränkt man sich bei der Checksummenberechnung auf den Header (16 Bits).

- *Source IP-Address*

Angabe der IP-Adresse des sendenden Rechners in hexadezimaler Darstellung (32 Bits)

HINWEIS

Nähere Angaben zur Adressierung im IP-Netzwerk enthält [Kapitel 3.](#)

- *Destination IP-Address*

Angabe der IP-Adresse des Empfangsrechners (32 Bits)

Das *Optionsfeld* (siehe [Abb. 2–9](#)) ist kein Pflichtbestandteil eines IP-Datagramms. Es besitzt keine feste Länge, sondern kann variabel lang sein und bis zu 40 Oktette umfassen. Das Feld nimmt vor allem Informationen zu Funktionen wie Routing, Diagnose oder Statistik auf.

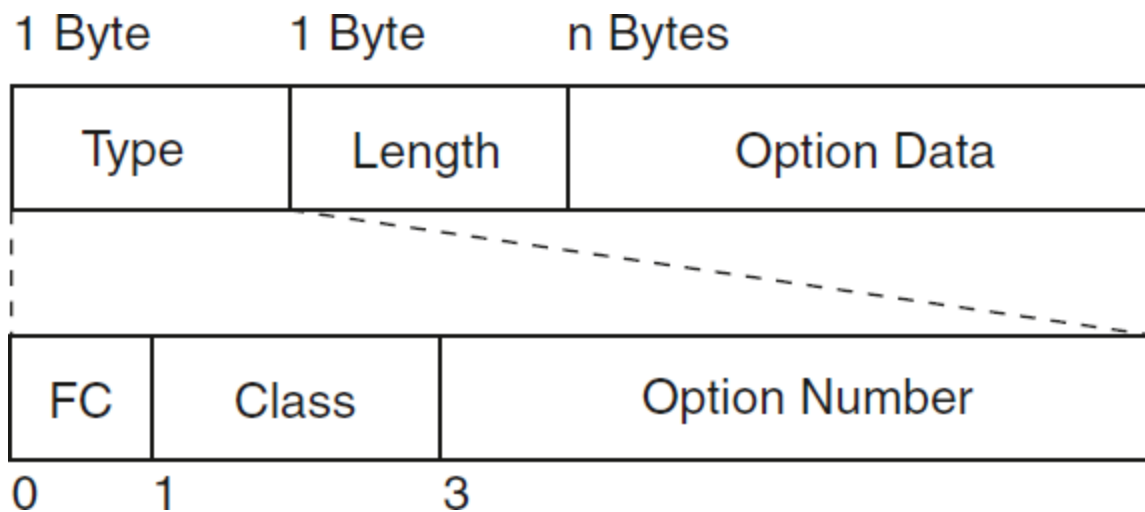


Abb. 2–9 Optionsfeld im IP-Datagramm

Das Optionsfeld unterteilt sich in drei Bereiche: Optionstyp, Optionslänge und Optionsdaten. Das Typenfeld wird folgendermaßen interpretiert:

- *FC*
»Flag-Copy«-Feld; gibt an, ob das Optionsfeld im Falle einer Fragmentierung kopiert werden soll (Wert »1«) oder nicht (Wert »0«).
- *Class*
Folgende Klassen sind verfügbar:

00: control

01: reserviert, wird zurzeit nicht benutzt

10: debugging und measurement

11: reserviert, wird zurzeit nicht benutzt

- *Option Number*

Verfügbare Optionen sind:

0: Ende der Optionenliste

1: keine Aktivitäten

2: Informationen (gemäß Sicherheitsaspekten DoD)

3: Loose Source Routing (gemäß Informationen des Senderechners)

4: Time Stamp

7: Routen-Trace

8: Stream-ID

9: Strict Source Routing

Da die verschiedenen Optionstypen entscheidende Bedeutung haben, werden diese nachfolgend ausführlich dargestellt.

- *Optionstyp 0* – Wert: 0 00 00000

Erhält der Optionstyp den Wert »0«, so wird dadurch das Ende der Optionenliste angezeigt. Das Längenfeld wird hier allerdings nicht belegt.

- *Optionstyp 1* – Wert: 0 00 00001

Dieser Optionstyp bewirkt, dass eine Liste mehrerer Optionen am Anfang eines 4-Byte-IP-Oktetts beginnen kann. Auf eine Operation bzw. Aktivität wird hier nicht hingewiesen. Das Längenfeld fehlt hier ebenfalls.

- *Optionstyp 130* – Wert: 1 00 00010

Die Gesamtlänge des Optionsfelds beträgt 11 Bytes. Die ersten beiden Bytes werden vom Optionstyp (1 Byte) und dem Längenfeld (1 Byte) belegt. Die nächsten zwei Bytes nehmen die Sicherheitsstufe auf. Dieses Feld wird auch SSS-Feld genannt. So wird beispielsweise die Sicherheitsstufe »vertraulich« durch die Bit-Sequenz <11110001 00110101> abgebildet. Andere Sicherheitsstufen sind:

»eingeschränkt« 10101111 00010011

»secret« 11010111 10001000

»top secret« 01101011 11000101

Das nächste Feld dieses Optionstyps ist das CCC-Feld (zwei Bytes). Es steht für die Angabe der Abteilung (*Compartment*). Das folgende HHH-Feld (zwei Bytes) beinhaltet die Handling Restrictions (gemäß Defence Intelligence Agency Manual DIAM 65-19). Das TCC-Feld (drei Bytes) steht für Transmission Control Code und ermöglicht die Kontrolle bzw. Steuerung des Datenverkehrs zwischen zwei Benutzern oder Gruppen von Benutzern.

- *Optionstyp 131* – Wert: 1 00 00011

In diesem Optionstyp werden Informationen zum *Loose Source Routing* hinterlegt. Der variable Aufbau dieser Feldgruppe ist in [Abb. 2-10](#) dargestellt.

1	0	0	0	0	0	1	1	Länge	Pointer	Routingdaten
---	---	---	---	---	---	---	---	-------	---------	--------------

Abb. 2-10 Optionstyp 68 – *Loose Source Routing*

Das Längensfeld umfasst die Gesamtlänge des Optionsfelds. Das 4-Byte-Pointer-Feld nimmt den Namen (IP-Adresse) des nächsten Routers auf, den das Datagramm erreichen soll. Bei jedem Passieren eines Routers wird dieser Eintrag entsprechend aktualisiert. Die Routing-Daten enthalten eine Liste von IP-Adressen der Router, die das vom Sender ausgelöste Datagramm passieren soll. Die letzte Adresse dieser Liste stimmt mit der Destination-IP-Adresse überein. Auf seinem Weg durch das Netzwerk wird somit durch dieses Verfahren die vollständige Route eines Datagramms dokumentiert (»Record Route«). Die Flexibilität dieses Ablaufs liegt in der Option, auch andere Router benutzen zu können, die in der Adressliste nicht aufgeführt sind. Beim *Strict Routing* (Optionstyp 137) besteht diese Möglichkeit nicht. Die zu Beginn vom Sender generierte Adressliste muss strikt eingehalten werden.

- *Optionstyp 7* – Wert: 0 00 00111

Auch dieses Optionsfeld setzt sich aus den Feldern Optionstyp, Länge, Pointer und Router-Daten zusammen. Hier erfolgt der sukzessive Aufbau einer Route, indem das Datagramm vom Sender zum nächsterreichbaren Router geschickt wird. Dort erfolgt ein entsprechender Eintrag im optionalen Router-Datenfeld. Im nächsten Schritt überträgt dieser Router das Datagramm an seinen unmittelbaren Router-Nachbarn. Dieser und jeder weitere Router wiederholt das Verfahren, bis der Zielknoten erreicht ist. Bei dieser Vorgehensweise muss vom Sender ein ausreichend großes Router-Datenfeld generiert werden, damit sämtliche Router ihre IP-Adressen in dieses Feld eintragen können. Ist es zu klein dimensioniert, so werden die Adressen der letzten Router bis zum Ziel nicht mehr protokolliert, und eine Aussage über die verwendete (vollständige) Route ist nicht möglich.

HINWEIS

Dieser Optionstyp findet primär Verwendung bei der Fehlerbehebung im Routing. Die entsprechende Umsetzung dieser Logik wird durch den in fast allen TCP/IP-

Implementierungen	verfügbaren	TRACEROUTE-Befehl
realisiert.		

- *Optionstyp 136* – Wert: 1 00 01000

Die Länge dieses Optionsfelds umfasst 4 Bytes. In den ersten beiden Bytes finden sich Optionstyp und Länge. Das dritte und vierte Byte nehmen den Stream-Identifizierer auf, der die höheren Protokollschichten über den zu verwendenden Transportmodus (Stream-Modus oder Datagramm-Modus) informiert. Beim Stream-Modus muss der Router dafür sorgen, dass ausreichend Ressourcen zur Verfügung stehen, um einen möglichst verzögerungsfreien Transport zu gewährleisten (Einsatz: Satellitenstrecken).

- *Optionstyp 68* – Wert: 0 10 00100

Für analytische Netzwerkbetrachtungen ist diese Option besonders bedeutsam. Sie ermöglicht die Aussage, wann ein Datagramm von einem Router übertragen bzw. verarbeitet wurde. Der Aufbau dieses Optionstyps ist in [Abb. 2-11](#) dargestellt. Das Längenfeld bezieht sich auf die Gesamtlänge des Optionsfelds. Der Pointer zeigt auf das nächste freie Byte, in dem ein Time-Stamp-Eintrag vorgenommen werden soll. Für den Fall, dass nicht alle Router ihren Time Stamp im Optionsfeld unterbringen konnten, wird hier ihre Anzahl eingetragen. Das Flag-Feld beinhaltet Time-Stamp-Optionen:

- 0: nur Time Stamps
- 1: IP-Adresse und zugehöriger Time Stamp
- 2: Vom Sender vordefinierte IP-Adress-Einträge werden von den Routern »ausgefüllt«.

01000100	Länge	Pointer	oflw	flg
IP-Adresse				
Time Stamp				
IP-Adresse				
Time Stamp				
...				

Abb. 2–11 Optionstyp 68 – Time Stamp

Das *Padding-Feld* schließt den IP-Frame (vor dem Datenteil) ab. Hier werden binäre Nullen als Filler benutzt, um einen IP-Frame stets auf einen 32-Bit-Block zu ergänzen.

Da das IP auf verschiedenen Netzwerktopologien eingesetzt werden kann, muss auf die Charakteristika der Netzwerke weitgehend Rücksicht genommen werden. So ist es

beispielsweise von entscheidender Bedeutung, die (hardwareabhängige) maximale Größe von IP-Datagrammen auf den verschiedenen Topologien mit ihren physikalischen Medien zu kennen. Die sog. *MTUs* (*Maximum Transfer Unit*) weichen in den einzelnen Netzwerken zum Teil erheblich voneinander ab, wie in der nachfolgenden Tabelle ersichtlich:

Netzwerktyp	Umfang
Token Ring (16 Mbit/s)	17914 Bytes
Token Ring (4 Mbit/s)	4464 Bytes
Ethernet	1500 Bytes
IEEE 802.3	1492 Bytes
X.25	576 Bytes

Die Größe eines Datagramms muss strikt nach den jeweils gültigen MTUs ausgerichtet werden. Um dies zu realisieren, ist eine Splittung der Originaldatagramme in kleinere Einheiten, die sogenannte *Fragmentierung*, und entsprechend ein Zusammenfügen der so entstandenen Datagrammfragmente zu dem Originaldatagramm (*Reassemblieren*) unbedingt erforderlich.

Ein unfragmentiertes Datagramm enthält im dritten Bit (*More-Fragment-Flag*) seines FLAG-Felds im IP-Header den Wert 0. Der FRAGMENT-OFFSET steht ebenfalls auf 0. Soll nun eine

Fragmentierung durchgeführt werden, muss im *Don't-fragment-Bit* des FLAG-Felds geprüft werden, ob eine Fragmentierung überhaupt durchgeführt werden darf. Ist dies nicht der Fall (Wert = 1), so wird das Datagramm nicht berücksichtigt, andernfalls wird der Fragmentierungsprozess fortgesetzt.

Gemäß MTU-Wert wird dann der Datenteil des IP-Datagramms in zwei oder mehrere Teildatagramme aufgeteilt, wobei jeder Datenteil eine Größe von acht Bytes oder ein Vielfaches dieses Wertes haben muss. Eine Ausnahme bildet lediglich das letzte Datagramm, dessen Datenteil auch kleiner sein kann. Die so aufgeteilten Datensegmente werden in ein vollständiges IP-Datagramm integriert. Die neuen IP-Header sind Kopien des Originaldatagramm-Headers mit einigen Modifikationen:

- Das MF-Bit eines jeden Datagramms erhält den Wert »1«; das letzte Datagramm jedoch »0«.
- Das Fragment-Offset-Feld wird mit dem Positionswert versehen, der auf die Stelle verweist, die von diesem Datenteil im Originaldatenteil des unfragmentierten Datagramms eingenommen wird.
- Sollten Optionsinformationen im Originaldatagramm enthalten sein, so entscheidet der jeweilige Optionstyp (fc-Bit

zu Beginn des Optionstypenfelds) über eine Weitergabe an die fragmentierten Datagramme.

- Die Header-Checksumme muss neu berechnet werden.
- Der Wert für das Gesamtlängelfeld (Bit 16 bis 31 im IP-Header) muss gesetzt werden.

Die fragmentierten Datagramme werden durch das Netzwerk geleitet und eventuell nochmals fragmentiert, je nachdem, ob die neue Datagrammgröße die MTU eines Netzwerks, das durchquert werden muss, erneut übersteigt. Es liegt natürlich auf der Hand, dass eine mehrfach vorzunehmende Fragmentierung die Performance im Netzwerk äußerst negativ beeinflusst. Beim letzten Router oder im Ziel-Host erfolgt schließlich das *Reassembly*, und die fragmentierten Datagramme werden zu dem Originaldatagramm zusammengesetzt:

- Sobald das erste Datagrammfragment am Reassembly-Knoten ankommt, werden Buffer allokiert, um für das »Datagramm-Mosaik« genügend Platz zu schaffen.
- Es ist unerheblich, in welcher Reihenfolge die Fragmente ankommen. Sie werden sukzessive an die richtige Position des reservierten Buffers kopiert.
- Gleichzeitig mit Ankunft des ersten Datagrammfragments wird ein Timer gestartet. Läuft dieser ab, bevor das letzte

Datagrammfragment angekommen ist, so wird das ursprüngliche Datagramm nicht mehr berücksichtigt und ignoriert. Der Buffer wird dann wieder freigegeben. Dies geschieht allerdings auch dann, wenn ein Fragmentfehler entdeckt wurde.

- Die eindeutige Identifikation der einzelnen Datagrammfragmente erfolgt im IP-Header über das IDENTIFICATION-Feld (beginnend mit Bit 32).

HINWEIS

Es hat sich gezeigt, dass in Wide-Area-Netzwerken eine kleinere MTU von Vorteil ist, da hier wahrscheinlichere *Re-Transmits* mit kleineren Datagrammen wesentlich unproblematischer bzw. schneller abgewickelt werden können als mit großen Datagrammen. In lokalen Netzwerken (LAN) hingegen ist die Wahl einer größeren MTU günstiger, um die vorhandenen hohen Bandbreiten besser ausnutzen zu können. In den letzten Jahren unterstützen moderne Router jedoch auch sogenannte *Jumbo Frames*, die eine wesentlich größere MTU als die im Ethernet üblichen 1500 Bytes ermöglichen und somit den Durchsatz deutlich erhöhen. Dabei ist jedoch darauf zu achten, dass sämtliche beteiligte Komponenten auch Jumbo-Frame-fähig sind.

Optional kann die hier beschriebene Fragmentierung und Reassemblierung von IP-Datagrammen variiert werden. Das entsprechende Verfahren wird *Path MTU Discovery* genannt. Dabei wird vorab mit dem Partnerrechner Kontakt aufgenommen, um die kleinste MTU zu ermitteln, die auf dem Weg zum Zielknoten angetroffen werden wird. Zur Vermeidung von Fragmentierung wird nun die so ermittelte kleinste MTU für den eigenen Rechner bzw. die Datagramme verwendet.

Da im IP-Datagrammverkehr bei Fehlern in der Kommunikation kein Recovery-Algorithmus vorgesehen ist, hat diese Aufgabe die nächsthöhere Protokollschicht zu übernehmen. Das TCP-Protokoll erfüllt diese Anforderung, UDP als ebenfalls höheres Protokoll allerdings nicht. Dies kann natürlich dann zu erheblichen Problemen führen, wenn ein gesicherter Datenfluss gefordert ist.

Internet Control Message Protocol (ICMP)

ICMP (*Internet Control Message Protocol*) ist ein IP-basiertes Protokoll, das für die Übermittlung von Fehlermeldungen verantwortlich ist, wenn beispielsweise ein Ziel-Host (*Destination Host*) oder ein Router mit einem Quell-Host (*Source Host*) kommuniziert und während dieser Kommunikation Fehler auftreten. Es benutzt IP, als wäre es selbst ein höheres

Protokoll, obwohl ICMP integraler Bestandteil des IP ist. Auch wenn ICMP für Fehlermeldungen zuständig ist, wird dadurch der IP-Datagrammverkehr nicht sicherer. Die Sicherung muss nach wie vor durch Protokolle wie das TCP übernommen werden. Um beim Melden von Fehlern außerordentlich große Datenmengen bzw. Datenfluss-Endlosschleifen zu vermeiden, werden Fehler im eigenen Protokoll (*ICMP Message Errors*) nicht gemeldet. Dies ist auch der Grund, warum bei fragmentierten IP-Datagrammen lediglich für das erste Fragment ggf. Fehlermeldungen generiert werden. Der ICMP-Frame wird stets mit einem vollständigen IP-Header versehen; die eigentliche ICMP-Message befindet sich im IP-Datenteil. Folgende ICMP-Message-Kategorien werden unterschieden:

- Fehlermeldungen
 - 3 Destination unreachable (Zielstation nicht erreichbar)
 - 4 Source quench (Buffer-Ressourcen verbraucht)
 - 5 Redirect (Pfadumleitung)
 - 11 Time exceeded (Timer abgelaufen)
 - 12 Parameter problem (Parameter-Problem)
- Informationsmeldungen
 - 0 Echo reply
 - 8 Echo request
 - 13 Time stamp

- 14Time stamp reply
- 15Information request
- 16Information reply
- 17Address mask request
- 18Address mask reply

Für ICMP-Messages wird im IP-Header das Type-of-Service-Feld auf »0« gesetzt und das Protocol-Feld erhält den Wert »1« als Protokollcode für ICMP. Der allgemeine Aufbau des ICMP-Message-Formats sieht also folgendermaßen aus:

Version	IHL	0	0	0	0	Total Length	
Identification						Flags	Fragment Offset
Time to Live		0	0	0	1	Header Checksum	
Source IP Address							
Destination IP Address							
Options/Padding							
ICMP-Type		ICMP-Code				ICMP-Checksum	
ICMP-Data							
...							

Abb. 2–12 *Aufbau einer ICMP-Meldung (Message)*

- *ICMP-Type* (8)

Die vorgenannten Ziffern in eckigen Klammern geben den Message-Typ an.

- *ICMP-Code* (8)

In Abhängigkeit vom Message-Typ erfolgt durch den ICMP-Code eine weitere Differenzierung der ICMP-Message.

- *ICMP-Checksum* (16)

Checksumme, beginnend mit dem Feld *Type*

- *ICMP-Data* (V)

Das ICMP-Datenfeld hat eine variable Länge. Normalerweise wird hier der IP-Header desjenigen IP-Datagramms einschließlich der ersten 64 Bit IP-Daten wiederholt, zu dem die ICMP-Message generiert wurde.

Im Folgenden werden die einzelnen *ICMP-Messages* (Meldungen) beschrieben und ihr Datagrammaufbau dargestellt:

- *DESTINATION UNREACHABLE*

Ein Router generiert diese Message, weil es ihm nicht möglich ist, sein Datagramm – gemäß Eintrag in seiner Routing-Tabelle – an die IP-Zieladresse (*destination address*) weiterzuleiten. Der Ziel-Host generiert diese Meldung, weil auf seinem Rechner die im IP-Datagramm spezifizierte Protokollnummer nicht aktiv ist.

ICMP-Code:

- 0 Netz nicht erreichbar
- 1 Host nicht erreichbar
- 2 Protokoll nicht erreichbar
- 3 Port nicht erreichbar

- 4 Fragmentierung erforderlich, allerdings Don't-fragment-Bit gesetzt
 - 5 Source Route nicht erreichbar
- *SOURCE QUENCH*

Diese Meldung wird durch einen Router generiert, weil er keine Buffer in ausreichendem Umfang zur Verfügung stellen kann, um eingehende Datagramme vor Weitergabe an das benachbarte Netzwerk zwischenspeichern. Ein Ziel-Host generiert diese *Message*, wenn die Datagramme in zu kurzen Zeitintervallen eingeht, sodass er (bzw. sein Netzwerkcontroller) diese nicht verarbeiten kann.

ICMP-Code: immer »0«
 - *REDIRECT*

Ein Router generiert diese *Message* (und sendet sie zum *Source-Host*), weil er eine günstigere Route erkannt hat und diese verwenden will. Die Adresse des neuen Routers trägt er anschließend in das Feld *Router IP-Adresse* ein. Sollte das IP-Datagramm jedoch einen Source-Routing-Eintrag enthalten, so wird diese ICMP-Message nicht generiert.

ICMP-Code:

 - 0 Umleitung erfolgt zu dem angegebenen Netzwerk.
 - 1 Umleitung erfolgt an den angegebenen Host.
 - 2 Umleitung erfolgt zu dem angegebenen Netzwerk mit dem *Type of Service*.

- 3 Umleitung erfolgt an den angegebenen Host mit dem *Type of Service*.

HINWEIS

Die ICMP-Codes 0 und 1 werden vom Routing-Protokoll RIP (*Routing Information Protocol*), die ICMP-Codes 2 und 3 vom Routing-Protokoll OSPF (*Open Shortest Path First*) ausgewertet. Näheres dazu enthält [Kapitel 4](#).

- *TIME EXCEEDED*

Ein Router generiert diese Message, wenn der »Time-to-live«-Wert oder der Reassembly-Timer abgelaufen ist.

ICMP-Code:

0 »Time-to-live«-Wert ist abgelaufen.

1 Reassembly-Timer ist abgelaufen.

- *PARAMETER PROBLEM*

Ein Router oder ein Ziel-Host generiert diese Message, wenn bei der Überprüfung des IP-Headers ein Fehler entdeckt wurde, der dazu führt, dass das Datagramm nicht bearbeitet werden kann. Das Feld *Pointer* (siehe [Abb. 2-13](#)) verweist auf die Byte-Position im Original-IP-Datagramm, an der der Fehler ermittelt wurde.

ICMP-Code: immer »0«

00001100 ICMP-Code ICMP-Checksum

Pointer unused (binary x"00")

IP-Header + 64 Bit IP-Data

Abb. 2–13 *ICMP-Datenfeld für ein Parameter-Problem*

- *ECHO REQUEST and REPLY*

Diese Meldung wird von einem Host generiert, wenn dieser überprüfen möchte, ob ein anderer Host im Netzwerk erreichbar ist. Das Feld *Identifier* wird initialisiert (*Sequence-Number-Feld* ebenfalls, wenn mehrere *Echo-Requests* gesendet werden) und dem Ziel-Host mit beliebigen Daten (56 bis max. 512 Bytes) zugeschickt. Dieser modifiziert das Typenfeld von 8 (*Echo Request*) auf 0 (*Echo Reply*) und schickt das Datagramm zurück. Dieser Erreichbarkeitstest wird durch den PING-Command implementiert.

ICMP-Code: immer »0«

- *TIME STAMP REQUEST and REPLY*

Time Stamp Requests and Replies werden verwendet, um zwei Hosts miteinander zeitlich zu synchronisieren bzw. um Performance-Messungen durchzuführen. Der sendende Host trägt zum Zeitpunkt des Versendens seinen Time Stamp in das Feld *Originate Time Stamp* ein. Der Ziel-Host füllt das Feld

Receive Time Stamp mit der Ankunftszeit des empfangenen Datagramms aus und trägt dann die Zeitdifferenz in das Feld *Transmit Time Stamp* ein (siehe [Abb. 2-14](#)).

ICMP-Code: immer »0«

bzw. 00001101 00001110	ICMP-Code	ICMP-Checksum
Originate Time Stamp		
Receive Time Stamp		
Transmit Time Stamp		

Abb. 2-14 *ICMP-Datenfeld für ICMP-Typ: Time Stamp Request and Reply*

- *INFORMATION REQUEST and REPLY*

Diese Meldung wird generiert, wenn ein Host die IP-Adresse des Netzwerks ermitteln möchte, in dem er sich befindet. Die IP-Zieladresse (*destination address*) wird auf »0« gesetzt (bedeutet: »dieses Netzwerk«) und es wird auf Antwort eines autorisierten Rechners gewartet. Die Antwort beinhaltet schließlich die korrekten Adressen in den Destination- und Source-Adressfeldern des IP-Headers.

ICMP-Code: immer »0«

HINWEIS

Dieser spezielle Optionstyp ist mittlerweile durch das RARP-Protokoll (*Reverse Address Resolution Protocol*) abgelöst worden.

- *ADDRESS MASK REQUEST and REPLY*

Ein Host generiert diese *Message*, wenn er von seinem Router die aktuelle Subnetzmaske seines Netzwerks in Erfahrung bringen möchte. In der Antwort ist das Feld *Subnet address mask* (siehe [Abb. 2-15](#)) entsprechend gefüllt. Die Anfrage (*Request*) kann direkt zu einem bestimmten Router geleitet (IP-Adresse des Routers ist bekannt) oder andererseits über Broadcasts (Rundsendungen an alle Netzteilnehmer) versendet werden.

ICMP-Code: immer »0«

bzw. 00010001 00010010	ICMP-Code	ICMP-Checksum
Identifier		Sequence Number
Subnet address mask		

Abb. 2–15 *Datenfeld für ICMP-Typ: Address Mask Request and Reply*

Address Resolution Protocol (ARP)

Alle Systeme (Rechner usw.), die in einem Netzwerk miteinander kommunizieren möchten, werden mit ihrer physischen Adresse angesprochen; dies ist in der Regel eine auf dem Netzwerkcontroller als Firmware permanent abgespeicherte Adresse (MAC-Adresse). Für die Kontaktaufnahme mit einem Partner wird diese allerdings nicht benutzt, sondern es wird eine logische Adresse verwendet, die für den Aufbau leistungsfähiger Adressstrukturen wesentlich besser geeignet ist. So lässt sich beispielsweise mit dem IP-Adresskonzept (siehe [Kap. 3](#)) ein logisches Netzwerk gezielt realisieren. Allerdings ist es für den jeweils eingesetzten Netzwerktreiber nicht möglich, einen Rechner über seine logische Adresse anzusprechen. Sie steht in keinerlei Verbindung mit der bereits erwähnten

Hardwareadresse des Controllers. Um eine solche Adressierung zu ermöglichen, muss die normalerweise verwendete logische Adresse in die Hardwareadresse umgesetzt werden. Diese Leistung erbringt das ARP (*Address Resolution Protocol*).

Eine sogenannte ARP-Adresstabelle (auch *ARP-Cache* genannt) dient als Informationsquelle für den Auflösungsprozess der logischen Adressen. Ist jedoch die gewünschte Adressinformation hier nicht verfügbar, so wird ein Broadcast generiert, der als ARP-Request von allen im Netz befindlichen Rechnern empfangen wird. Erkennt ein Rechner in dem empfangenen ARP-Request seine eigene logische Adresse, so schickt er dem anfragenden Rechner seine physische Adresse in einem ARP-Reply zurück. Dieser übernimmt nun die neue Adressinformation (neben eventuell vorhandenen *Source-Routing*-Einträgen) in seinen ARP-Cache und kann zukünftige Datagramme an seinen Partnerrechner direkt übermitteln. Zur Vermeidung unnötiger Broadcasts trägt auch der Ziel-Host die entsprechenden Adressinformationen seines Partners in den eigenen Cache ein.

HINWEIS

ARP-Datagramme werden nicht über Router-Grenzen hinweg übertragen. Die Beantwortung von ARP-Requests führt der

Router selbst durch und leitet alle Datagramme entsprechend seinen Routing-Einträgen an die Zielrechner weiter. ARP-bedingte Broadcasts gelangen somit nicht in alle theoretisch erreichbaren Netzwerke, sondern bleiben im lokalen Router-Netzwerk. Werden zwei Netzwerke allerdings über Brücken oder Switches (Layer 2) zu einem logischen Netzwerk miteinander verbunden, so besteht die Gefahr, dass ARP-Broadcasts regelrechte »Broadcast-Stürme« auslösen und diese Netzwerke mit Requests überfluten.

Der ARP-Frame besteht aus dem *MAC-Header* (je nach Netzwerkstandard) und dem *ARP-Packet (Request/Reply)*. Der Aufbau eines MAC-Headers wurde bereits erläutert. An dieser Stelle soll das *ARP-Packet*, so wie in [Abbildung 2-16](#) dargestellt, erläutert werden.

Hardwaretyp		Protokoll-Typ
Länge der physischen Adresse	Länge der Protokolladresse	Operation Code
Physische Quelladresse		
Physische Quelladresse	Quell-Protokolladresse	
(Fortsetzung)		
Quell-Protokolladresse	Physische Zieladresse	
(Fortsetzung)		

Physische Zieladresse (Fortsetzung)
Ziel-Protokolladresse

Abb. 2–16 *ARP-Packet-Format (hier ARP-Request)*

- *Hardwaretyp* (16)
Typ des physikalischen Netzes (z.B. 1 = Ethernet, 6 = IEEE-802-Protokolle usw.)
- *Protokolltyp* (16)
Angabe der Protokollnummer (gemäß Feld *Ethertype* im Ethernet-Header). Der Protokolltyp für IP lautet demnach 0x0800.
- *Länge physische Adresse* (8)
Angabe der Länge (in Bytes) der physischen Adresse. In der Regel wird hier der Wert »6« eingetragen, da dies die Länge der MAC-Adresse ist (gemäß IEEE-802-Standard).
- *Länge Protokolladresse* (8)
Hier wird die Länge der Protokolladresse eingetragen. Für IP wäre das beispielsweise der Wert »4«, da eine IP-Adresse aus vier Oktetten besteht.
- *Operationscode* (16)
Spezifikation, ob es sich um einen ARP-Request (1), ARP-Reply (2), RARP-Request (3) oder RARP-Reply handelt (RARP = *Reverse Address Resolution Protocol*; siehe [Abschnitt 2.3.4](#))

- *Physische Quelladresse* (48)
Gemäß IEEE 802: MAC-Quelladresse
- *Quell-Protokolladresse* (32)
Quell-IP-Adresse
- *Physische Zieladresse* (48)
Gemäß IEEE 802: MAC-Zieladresse (wird beim ARP-Request mit binär »0« initialisiert; die physische Adresse des Ziel-Hosts ist ja unbekannt)
- *Ziel-Protokolladresse* (32)
Ziel-IP-Adresse

HINWEIS

Die Gesamtlänge des »ARP-Packets« beträgt somit 28 Bytes. Dabei besitzt der ARP-Reply den gleichen Aufbau wie der ARP-Request. Allerdings lautet der Wert für den Operationscode »2« statt »1«, und die physische Quelladresse (ab Byte 8) wird durch die ermittelte (bzw. angefragte) physische Adresse ersetzt.

Reverse Address Resolution Protocol (RARP)

RARP arbeitet genau in umgekehrter Richtung zum ARP-Protokoll. Hier ist die IP-Adresse nicht bekannt, und die physische Adresse soll gesucht werden. Dies geschieht dadurch,

dass ein Host einen *RARP-Request* aussendet und dabei seine physische Adresse angibt.

Im Netzwerk installierte RARP-Server sorgen nach Durchsicht eigener Referenztabellen für die Überführung der physischen Adresse in die IP-Adresse. Anschließend wird diese Information als RARP-Reply an den sendenden Host zurückgeschickt. Dieser übernimmt den zuerst empfangenen Reply und ignoriert alle folgenden. Das RARP-Packet-Format ist mit dem ARP-Packet-Format identisch. RARP kommt oft dort zum Einsatz, wo *Diskless Stations* eingesetzt werden, also Rechner, die über kein Speichermedium verfügen, wo sie ihre IP-Adresse hinterlegen und jederzeit abrufen können. Sie »booten« also lediglich mit Kenntnis ihrer physischen Netzwerkadresse; die IP-Adresse ist hingegen unbekannt.

Routing-Protokolle

Die Zusammenführung von unterschiedlichen Netzwerken, die geografischen, topologischen oder organisatorischen Gesetzmäßigkeiten unterliegen, werden – sofern Routing-Protokolle eingesetzt sind – durch Router realisiert. Sie bilden gewissermaßen sowohl die physischen Endpunkte des lokalen Netzwerks als auch seine Verbindungsknoten zu benachbarten Netzen.

Bedingt eben durch die netzwerkübergreifende Kommunikation ist es notwendig, leistungsfähige Steuerungsmechanismen einzusetzen, die einen schnellen und sicheren Datenfluss gewährleisten können. Zu diesem Zweck wurden neben speziell optimierter Hardware, also den dedizierten Routern, explizite Routing-Protokolle entwickelt, die den genannten Anforderungen entsprechen. Sie arbeiten alle nach unterschiedlichen Prinzipien und werden hinsichtlich ihrer Leistungsstärke auch dementsprechend beurteilt bzw. eingesetzt. Es sind dies im Einzelnen:

- Routing Information Protocol (RIP; RIP-2)
- Hello Protocol
- Exterior Gateway Protocol (EGP)
- Gated
- Open Shortest Path First (OSPF)
- IS-IS-Routing
- Border Gateway Protocol (BGP)

Daneben existieren weitere proprietäre Lösungen bestimmter Hersteller wie beispielsweise IGRP (*Interior Gateway Routing Protocol*) oder EIGRP (*Enhanced Interior Gateway Routing Protocol*); auf diese speziellen Protokolle wird jedoch im Rahmen unseres Buches nicht näher eingegangen.

HINWEIS

Nähere Angaben zum Thema Router und zu den diversen Routing-Protokollen enthält [Kapitel 4](#).

Protokolle der Transportschicht (Layer 4)

Das Internet Protocol (IP) besitzt keinen Mechanismus zur Gewähr von Datensicherheit oder Datenflusssteuerung; dies muss somit Aufgabe höherer Protokollschichten sein. So sorgen Protokolle auf der Transportschicht (Layer 4; siehe [Abb. 2–17](#)) unter anderem dafür, dass

- über dedizierte Transportverbindungen Daten übertragen werden können,
- der Auf- und Abbau von Verbindungen kontrolliert erfolgt,
- eine Kontrolle, Fehlererkennung und Flusssteuerung über die *gesamte* Verbindung ermöglicht wird,
- mehrere Verbindungen gleichzeitig zu *einem* Rechner verwaltet werden können (Multiplexing),
- optimierter Datenfluss (Windowing) ermöglicht wird und
- eine Priorisierung im Datentransport (Urgent-/Push-Mechanismus) realisiert werden kann.

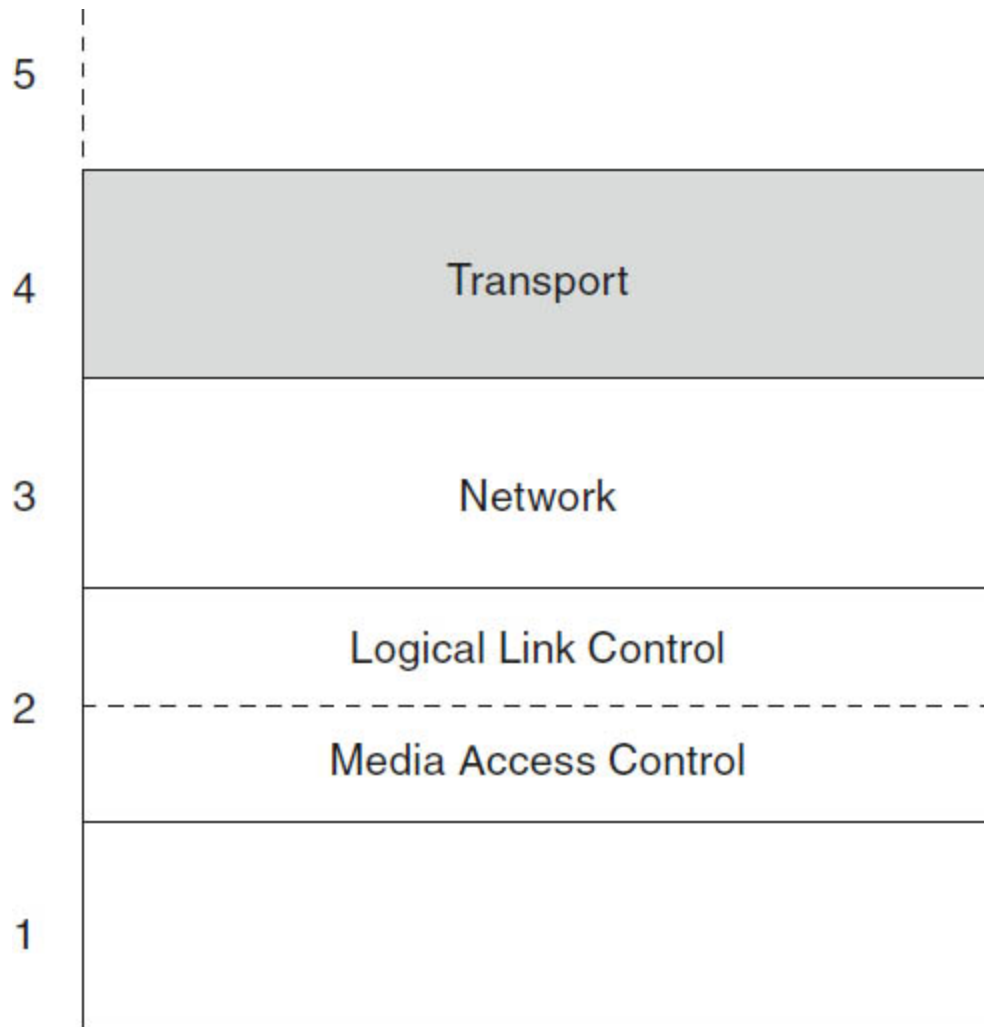


Abb. 2–17 Layer-4-Schicht (Transport)

Die genannten Anforderungen werden durch das *Transmission Control Protocol* (TCP) erfüllt. Im Gegensatz zum IP stellt es ein verbindungsorientiertes Protokoll dar und arbeitet aus Sicht der Anwendungen, die es bedient, datenstromorientiert. Anwendungen brauchen sich nicht um Segmentierung oder Blockung zu kümmern. Für sie existiert lediglich ein kontinuierlicher, byteweiser Datenstrom. Dies verwundert umso mehr, als beim Blick auf das zweite markante Protokoll

der Transportschicht (*User Datagram Protocol* = UDP) auffällt, dass fast alle genannten Merkmale auf dieses Protokoll nicht zutreffen. UDP rechtfertigt seine Zugehörigkeit zum Layer 4 eigentlich nur durch seine Funktion des Datentransports. Es ist ungesichert, verbindungslos und auch ohne Recovery- oder Fehlermanagement.

Im Folgenden werden die einzelnen Leistungsmerkmale des TCP detailliert beschrieben und seine Bedeutung innerhalb des Schichtenmodells herausgearbeitet, wobei insbesondere die Themen Verbindungsmanagement, *Three Way Handshake*, Datenfluss, *Windowing* und *Retransmission* im Vordergrund stehen. Vor diesen Erläuterungen soll an dieser Stelle kurz auf die beiden Begriffe *Port* und *Socket* eingegangen werden.

Ausgehend von der deutschen Übersetzung des Worts *Port* mit »Ziel«, »Tor« oder »Hafen« lässt sich rasch sein Verwendungszweck im TCP/IP-Umfeld erfassen: Der Port repräsentiert eine 16-Bit-Adresse, die zur Identifikation eines eindeutigen Zugangspunkts dient, den das TCP- oder UDP-Protokoll benötigt, um mit den übergeordneten Anwendungen bzw. Anwendungsprotokollen (z.B. FTP oder TELNET) Daten austauschen zu können. Man unterscheidet zum einen die sogenannten *well known ports*, die bereits seit Langem definiert sind und mittlerweile weltweit Verwendung finden (siehe [Abb.](#)

2–18). Sie umfassen einen Bereich von 0 bis 1023 und werden von der *Internet Assigned Numbers Authority* (IANA) vergeben. Alle übrigen Ports im Bereich zwischen 1024 und 65.535 unterstehen nicht der Kontrolle der IANA. Sie stehen – ebenfalls weltweit – den Programmierern für ihre eigenen Entwicklungen zur Verfügung.

111	161	69	25	21	23
SUN RPC	SNMP	TFTP	SMTP	FTP	TELNET
User Datagram Protocol (UDP)			Transmission Control Protocol (TCP)		

Abb. 2–18 Beispiele sogenannter *well known ports*

Per Definition versteht man unter einem *Socket* den durch das Tripel »Protokoll«, »IP-Adresse« und »Port-Nummer« eindeutig bestimmbaren Endpunkt einer Kommunikation. In der Praxis wird durch den Begriff *Socket* allerdings eine Schnittstelle bezeichnet, die zwischen zwei Anwendungsprozessen auf einem Host eine Client-Server-Beziehung herstellt und nach festgelegten Regeln die im Layer 4 zur Verfügung gestellten Transportdienste nutzt. Es werden grundsätzlich drei Socket-Typen unterschieden:

- *Stream Sockets*

Sie greifen auf den Transportdienst des TCP-Protokolls zu, wobei der stattfindende Datenfluss auf einer sicheren Verbindung zwischen zwei Sockets realisiert wird.

- *Datagram Sockets*

Analog erfolgt hier der Transport von Daten nach den Gesetzmäßigkeiten des UDP-Protokolls: schnell und weniger zuverlässig.

- *Raw Sockets*

Dieser Typ ermöglicht den direkten Zugriff auf die Netzwerkschicht (Internet Protocol).

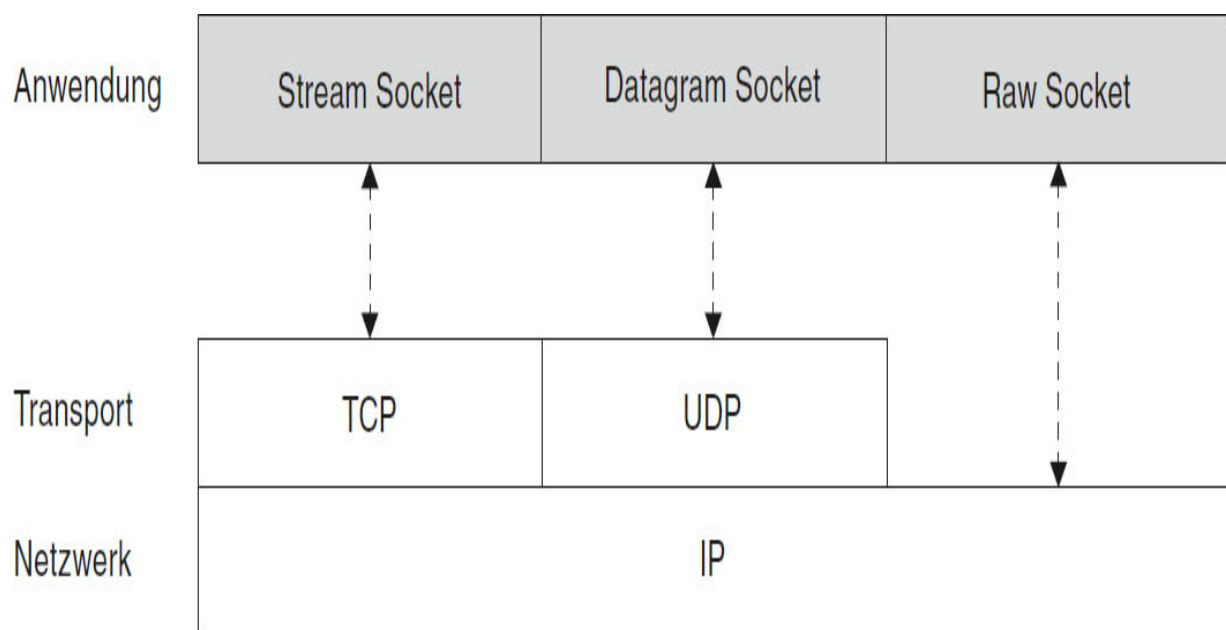


Abb. 2–19 Protokollzugriff der verschiedenen Socket-Typen

Transmission Control Protocol (TCP)

Das *Transmission Control Protocol* wird als das zentrale Protokoll innerhalb der TCP/IP-Protokollfamilie betrachtet. Es bildet die Grundlage für die Leistungsfähigkeit zahlreicher Anwendungsprotokolle wie beispielsweise FTP (*File Transfer Protocol*), TELNET, SMTP (*Simple Mail Transfer Protocol*) oder das X-Protokoll für XWindows/Motif-Anwendungen. TCP lässt sich in seinen Hauptfunktionen folgendermaßen charakterisieren:

- Datenstromtransfer
- Virtuelle Full-Duplex-Verbindungen
- Datenflusssteuerung
- Fehlererkennung
- Prioritätensteuerung

Die grundlegende Funktionsweise lässt sich folgendermaßen beschreiben: Eine Anwendung veranlasst TCP zum Aufbau einer Verbindung und übergibt dann datenstromorientiert (in Oktetten) die Daten. Der mitunter sehr umfangreiche Datenstrom wird von TCP geteilt bzw. segmentiert und jedes Segment wird mit einem eigenen Header versehen, worauf in Layer 3 die Übergabe an das Internet Protocol erfolgt. Nach dem Transport der Daten über das Netzwerk erfolgt die

Datenübergabe an das TCP, nachdem IP die Header entfernt hat. TCP ist damit beauftragt, den ankommenden Datenfluss auf Fehler zu überprüfen, ankommende Segmente zu sortieren, TCP-Header zu entfernen und schließlich die Daten der Zielanwendung zuzuführen. Sobald die Anwendung entsprechende Anweisungen erteilt, wird die logische Verbindung durch TCP beendet.

Die wesentlichen Leistungsmerkmale des TCP-Protokolls sollen nachfolgend in einer ausführlichen Betrachtung erläutert werden. Dazu ist es erforderlich, zunächst das *TCP-Segment-Format* kennenzulernen, so wie in [Abb. 2-20](#) dargestellt.

Source Port					Destination Port				
Sequence Number									
Acknowledgement Number									
Data Offset	Reserved	U R G	A C K	P S H	R S T	S Y N	F I N	Window Size	
Checksum					Urgent Pointer				
Options/Padding									
Data									
...									

Abb. 2–20 *Aufbau des TCP-Segment-Formats*

- *Source Port* (16)
Dieses Feld beinhaltet den Quell-Port der TCP-Anwendung, die den Frame generiert hat.
- *Destination Port* (16)
Dieses Feld beinhaltet den Ziel-Port der TCP-Anwendung, die den Frame empfangen soll.
- *Sequence Number* (32)

Die *Sequence Number* führt eine Nummerierung des jeweils ersten Daten-Bytes bzw. der Daten-Oktette innerhalb des Gesamtdatenstroms durch. Besteht ein Datenstrom beispielsweise aus fünf TCP-Segmenten, wobei jedes Segment (in diesem Beispiel) 25 Daten-Oktette aufnehmen kann, so erhielte die *Sequence Number* des ersten Segments den Wert 1. Die Sequenznummern aller weiteren Segmente bekämen die Werte 26, 51, 76 und 101. Gerät nun während des Netzdurchlaufs über IP die Segmentreihenfolge durcheinander, so ist das TCP am Ziel-Host dennoch in der Lage, die Segmente wieder in die richtige Reihenfolge zu bringen.

Lediglich beim *Three Way Handshake* beginnt der Wert der Sequenznummer (*Initial Sequence Number* = ISN) mit einem Zufallswert, sodass das erste Daten-Byte auf Position $ISN + 1$ zu finden ist. TCP verwendet normalerweise variable Segment-Lebenszeiten, da die Verkehrszeit von Segmenten bzw. Datagrammen auf größeren Netzwerken starken Schwankungen unterliegen kann. Unterschiedliche Leitungsgeschwindigkeiten und eine divergierende Auslastung der Router im Netzwerk können dafür verantwortlich sein (dies ist im Rechnerverbund Internet häufig der Fall).

- *Acknowledgement Number* (32)

Alle empfangenen Daten werden dem Sender durch die *Acknowledgement Number* bestätigt. Sobald das ACK-Bit gesetzt ist, enthält dieses Feld den Wert der *Sequence Number*, die der Empfänger nun vom Sender erwartet.

- *Data Offset* (4)

Dieses Feld gibt die Länge des TCP-Headers in 32-Bit-Blöcken an. Erforderlich ist diese Angabe, da das Optionsfeld variabel lang sein kann.

- *Reserved* (6)

Dieses Feld wird zurzeit nicht benutzt und enthält binäre Nullen.

Control Flags

- *Urgent-Pointer-Flag* (1)

Durch dieses Bit werden Vorrangdaten gekennzeichnet. Ein TCP-Segment dieses Typs wird oft zur Unterbrechung von Prozessen benötigt.

- *Acknowledgement-Flag* (1)

Lautet der Wert dieses Flags 1, so wird mit diesem Segment u.a. der Empfang von Daten bestätigt.

- *Push-Flag* (1)

Der empfangende TCP-Layer wird informiert, dass ein derart markiertes Segment unmittelbar an die zu bedienende Anwendung weitergeleitet werden muss (normalerweise

werden Segmente erst in Buffern gesammelt und zur Vermeidung von Overhead in größeren »Portionen« weitergegeben). Im TEL-NET-Dialog ist dieses Bit permanent aktiv.

- *Reset-Flag* (1)

Hierdurch wird dem Empfänger mitgeteilt, dass die Verbindung aufgrund einer nicht näher bestimmbareren Fehlersituation beendet werden soll.

- *Synchronization-Flag* (1)

Der Sender teilt dem Empfänger mit, dass eine Verbindung aufgebaut werden soll, und leitet damit den Synchronisationsvorgang ein (Three Way Handshake).

- *Final-Flag* (1)

Durch dieses Bit wird das endgültige, ordentliche Verbindungsende eingeleitet. Sendet der Empfänger – nachdem der Sender dieses Bit bereits aktiviert hat – ebenfalls ein FIN, so ist die Verbindung beendet.

- *Window Size* (16)

Um einen »Überlauf« des Buffers beim Empfänger während einer Verbindung zu vermeiden, wird dem Sender über das *Windowing* in diesem Feld kontinuierlich mitgeteilt, wie viele Daten (in Bytes) er unbestätigt senden kann.

- *Checksum* (16)

Wert der Prüfsumme aus *TCP-Header* und *Pseudo-Header*. Der Pseudo-Header besteht aus der IP-Quelladresse, der IP-Zieladresse, einem 8-Byte-Leerfeld, dem Protokoll-Identifizierer und der Angabe über die Länge des TCP-Segments.

- *Urgent Pointer* (16)

Der Wert im Urgent-Pointer-Feld weist auf das Ende der Vorrangdaten innerhalb eines TCP-Segments hin. Er stellt einen Offset-Wert dar, der in Addition mit der Sequence Number die beschriebene Datenposition identifiziert.

- *Options* (variabel)

Das Optionsfeld besteht aus einem Optionstyp, einer Optionslänge (jeweils 1 Oktett) und ggf. aus den Optionsdaten. Zurzeit sind lediglich drei Optionstypen definiert:

0: Ende der Optionenliste

1: keine Aktivitäten

2: maximale Segmentgröße MSS

- *Optionstyp 0* – Wert 00000000

Enthält der Optionstyp den Wert »0«, so wird dadurch das Ende der Optionenliste angezeigt. Das Längensfeld wird hier allerdings nicht belegt.

- *Optionstyp 1* – Wert 00000001

Dieser Optionstyp wird zur Trennung zwischen zwei Optionen verwendet. Das Längensfeld fehlt hier ebenfalls.

- *Optionstyp 2* – Wert 00000010

Hier wird die maximale Segmentgröße zwischen zwei Kommunikationspartnern festgelegt. Das Längenfeld erhält den Wert »4« (»0100«). Diese Option wird lediglich beim Verbindungsaufbau verwendet.

HINWEIS

Das letzte Feld (*Padding*) wird so lange mit binären Nullen aufgefüllt, bis die Gesamtlänge des 32-Bit-Blocks erreicht ist.

Verbindungsmanagement

Das Verbindungsmanagement umfasst festgelegte Maßnahmen zu Verbindungsaufbau, Verbindungskontrolle und Verbindungsabbau. Die sogenannte *Connection Primitive* leitet den Verbindungsaufbau zwischen zwei Prozessen ein: Prozess A geht in den *Passive open*-Status; d.h., er wartet auf Kontaktaufnahme durch einen bestimmten Kommunikationspartner (*Specific Originator*) oder mehrere potenzielle Sender (*Any Originator*). Der (oder die) Kommunikationspartner initiieren dann einen *Connection Request* und gehen dadurch in den *Active open*-Status.

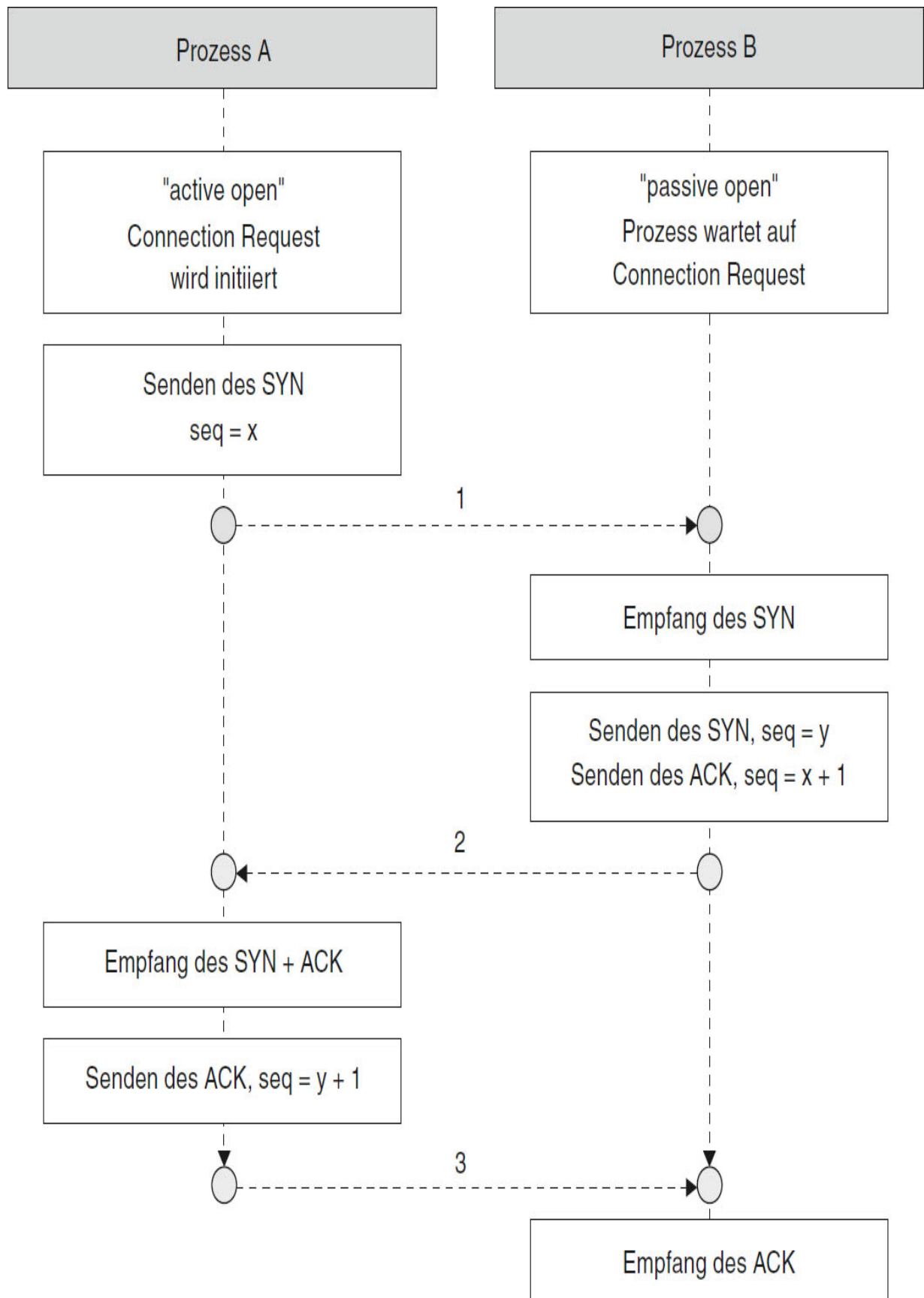


Abb. 2–21 *Prinzip des Three Way Handshakes*

Bis jedoch die Verbindung aufgebaut ist, müssen drei Phasen durchlaufen werden, wobei dieser Prozess auch als *Three Way Handshake* bezeichnet wird. In der *ersten* Phase erfolgt ein Connection Request vom Originator, indem er seinem Partner-Host ein Segment mit einem gesetzten SYN-Flag schickt. Diese Verbindungsanfrage beinhaltet bereits die IP-Adresse des Ziel-Hosts und den Port, über den eine Verbindung gewünscht wird. An dieser Stelle würde auch der bereits im letzten Abschnitt erwähnte ARP-Broadcast vorgenommen, wenn sich die angegebene IP-Adresse nicht im eigenen ARP-Cache befindet. Mit der vorhandenen oder ermittelten MAC-Adresse geht nun der IP-Frame ins Netz. Die *zweite* Phase beginnt beim Ziel-Host mit dem Empfang des aus der ersten Phase übermittelten SYN. Er quittiert diesen Empfang mit gesetztem SYN-Flag und fügt zusätzlich ein ACK-Flag als Bestätigung hinzu. Dadurch signalisiert er dem *Requesting Host* seine Bereitschaft, eine logische Verbindung einzugehen. Die *dritte* Phase umfasst den Eingang der Bestätigung des Ziel-Hosts beim Requesting Host und den Versand eines letzten ACK an den Ziel-Host (in dieser Aufbauphase der Verbindung) als weitere Bestätigung, dass nun die Verbindung aufgebaut werden kann. Der

Verbindungsaufbau wird nun vollzogen und es können Daten übertragen werden.

Die durch einen Einigungsprozess während dieser Aufbauphase ausgetauschte Sequenznummer zwischen Quell- und Ziel-Host (Sender teilt dem Empfänger seine aktuelle *Sequence Number* mit und der Empfänger gibt seine Sequenznummer analog an den Sender weiter) sorgt für die Initialisierung der Verbindung.

Nach dem Aufbau der Verbindung und der erfolgten Datenübertragung gibt es zwei Möglichkeiten, diese zu beenden: Im *Fehlerfall* wird ein *Abort primitive* eingeleitet, der die Verbindung durch Setzen des RST-(Reset-)Flags sofort beendet. Datenflussmechanismen zur Vermeidung von Datenverlust werden dabei nicht berücksichtigt. Eine *normale Beendigung* der Verbindung wird durch ein gegenseitiges Abstimmungsverfahren über das FIN-Flag vorgenommen. Daten gehen dabei nicht verloren.

Datenflusssteuerung und Windowing

Es ist bereits mehrfach darauf hingewiesen worden, dass eine der herausragendsten Eigenschaften höherer Protokolle wie TCP die Fähigkeit ist, für eine ausreichende Datensicherheit auf dem Transportweg zwischen zwei oder mehreren Hosts bzw.

ihren kommunizierenden Prozessen zu sorgen. Dieses Leistungsmerkmal ist auch unbedingt notwendig, da auf der Netzwerkschicht (IP) eine solche Fähigkeit fehlt.

In der Praxis bedeutet dies: Der Empfang eines TCP-Segments, das von Host A zu Host B gesendet wird, muss bestätigt werden (ACK = *Acknowledgement*), damit ein sicherer Datentransport zu jedem Zeitpunkt einer Kommunikation gewährleistet ist. Dies bedeutet weiterhin, dass ein zweites (und jedes weitere) Segment erst dann versendet werden kann, wenn die Quittung für das vorherige eingetroffen ist. Dieses Verfahren stellt zwar ein Maximum an Sicherheit dar, lässt allerdings den äußerst wichtigen Aspekt der Geschwindigkeit im Netzwerk (Zeitverhalten des Datenstroms) nahezu völlig unberücksichtigt. Um hier einen Kompromiss zu finden, wurde das Prinzip der *Sliding Windows* oder auch *Windowing* eingeführt. Nach diesem Verfahren erfolgt nicht nach jedem versendeten TCP-Segment eine entsprechende Bestätigung, sondern es wird lediglich eine Gruppe von Segmenten quittiert.

Der Empfang einer bestimmten Anzahl von Segmenten erfordert beim Empfänger einen entsprechend hohen Bedarf an Buffer. Da ein Host für den Betrieb einer logischen Verbindung einen bestimmten Speicherbereich allokiert und diesen nicht ohne Weiteres dynamisch vergrößern kann, muss

er damit haushalten. Die *Window-Size* stellt nun eine Limitierung der Datenmenge (in Bytes) dar, die vom Empfänger angenommen werden kann, ohne ein ACK zu schicken. Sobald seine *Window-Size* erreicht ist, muss ein ACK zum Sender erfolgen. Der Empfänger ist dann wieder in der Lage, gemäß der *Window-Size* neue Segmente in Empfang zu nehmen. Bei jedem Verbindungsaufbau wird die Größe der *Window-Size* zwischen Sender und Empfänger vereinbart. Sie kann allerdings im Laufe einer Verbindung dynamisch vom Empfänger – seinen Möglichkeiten entsprechend – angepasst werden. Bei eigenen Ressourcenproblemen kann die *Window-Size* sogar bis zu einem Wert »0« verringert werden. Dies bedeutet jedoch, dass der Datenfluss zum Stillstand kommt. Durch ein allmähliches Vergrößern dieser *Window-Size* durch den Empfänger wird der Datenfluss wieder erhöht. Eine neue *Window-Size* wird stets innerhalb eines ACK-Segments vom Empfänger festgelegt. Diese Datenflussteuerung soll anhand eines Beispiels erläutert werden:

Angenommen, die Segmentgröße wird auf 1000 Bytes festgelegt, die *Window-Size* auf 4000 Bytes. Der Sender schickt sein erstes Segment mit einer *Sequence Number* von 1. Es folgt ein zweites Segment (SEQ = 1001), ein drittes (SEQ = 2001) und ein viertes (SEQ = 3001). Die *Window-Size* ist nach viermaligem Senden eines 1000-Byte-Segments erreicht. Der Empfänger

muss nun ein ACK senden und damit den Empfang der Segmente 1 bis 4 bestätigen. Zur Reduzierung der Netzlast darf er sich auf den Versand eines ACK für das vierte Segment beschränken, um zu dokumentieren, dass er alle Segmente erhalten hat. Eine Bestätigung jedes einzelnen Segments ist nicht erforderlich. Hat er jedoch nur die ersten beiden Segmente erhalten, so bestätigt er lediglich mit einem ACK(2). Das fiktive »Fenster« gleitet nun von Segment 1 im ersten Byte zu Segment 5 auf Byte 4001, so wie in [Abb. 2-22](#) dargestellt. Es überdeckt lediglich diejenigen Oktette (bzw. Bytes), die bereits versendet, aber noch nicht bestätigt worden sind, sowie diejenigen Oktette, die noch nicht versendet wurden.

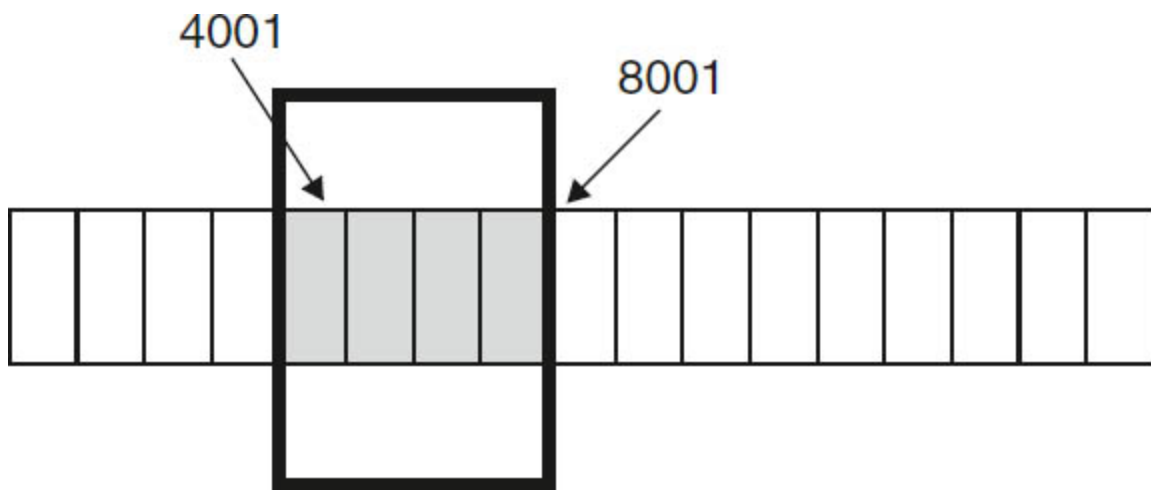


Abb. 2-22 *Darstellung eines Sliding Window*

Retransmission

Trifft die Bestätigung für ein gesendetes Segment innerhalb einer bestimmten Zeit nicht ein, so wird dieses Segment erneut übertragen. Das entsprechende Zeitintervall wird im *Round Trip Time-Timer* (RTT) hinterlegt und kontinuierlich neu berechnet.

Angenommen, im vorgenannten Beispiel geht Segment 3 verloren. Der Empfänger versendet ein ACK(2); d.h., Segment 1 und Segment 2 wurden ordnungsgemäß empfangen. Das *Sliding Window* schiebt sich um 2000 Oktette vor (siehe [Abb. 2-23](#)) und der Sender überträgt (wegen der Window-Size von 4000 Bytes) lediglich Segment 5 und 6. Anschließend muss er auf ein weiteres ACK warten, da er ja sonst die Window-Size überschreiten würde.

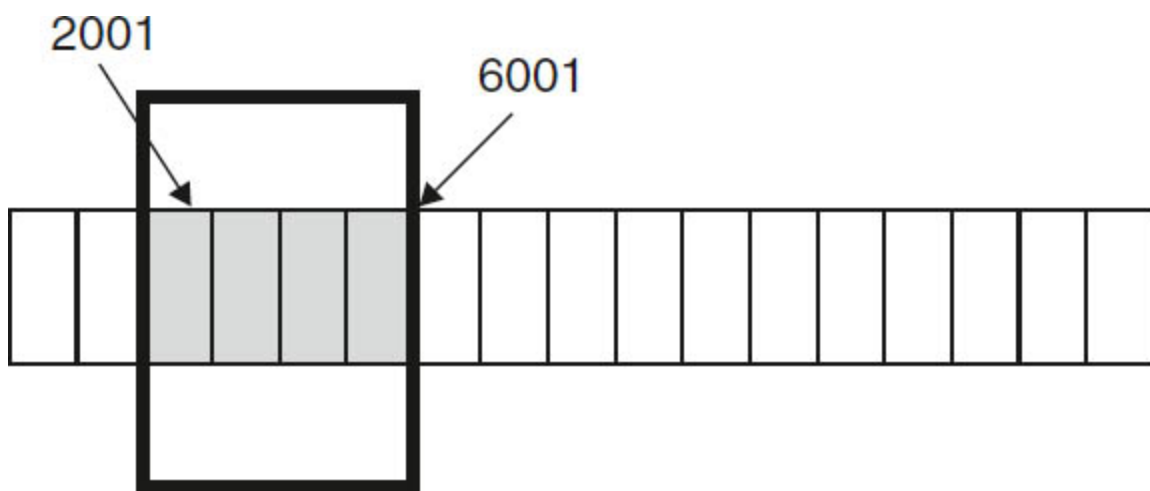


Abb. 2-23 ACK(2) verschiebt das Window um 2000 Oktette.

Mittlerweile ist allerdings der RTT für Segment 2 abgelaufen, und der Sender führt einen Retransmit für Segment 2, 3, 4 und 5 durch (für Segment 6 darf kein Retransmit erfolgen, da sonst erneut die Window-Size überschritten würde). Der Empfänger quittiert mit ACK(5). Nun gleitet das Window um 4000 Oktette weiter. Segment 6 ist noch »unterwegs«, da der RTT für dieses Segment noch nicht abgelaufen ist. ACK(6) trifft ebenfalls beim Sender ein, und das Window verschiebt sich um weitere 1000 Oktette, so wie in [Abb. 2-24](#) dargestellt.

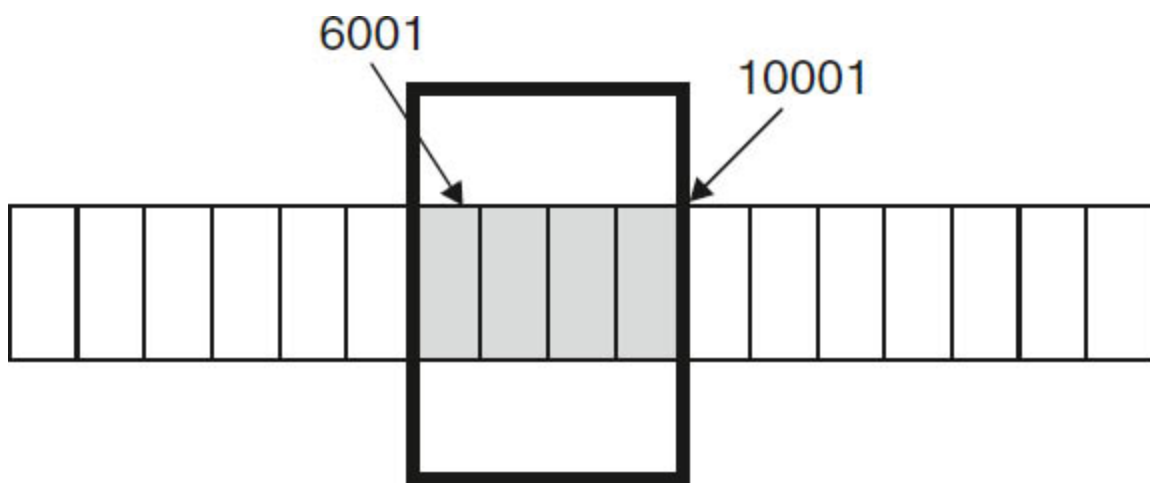


Abb. 2-24 *Darstellung des Window nach ACK(6)*

User Datagram Protocol (UDP)

IP besitzt zwar die Fähigkeit, Datagramme von Host A nach Host B zu versenden, es ist jedoch nicht in der Lage, diese an Anwendungen weiterzugeben. Dies jedoch vermag UDP. Außerdem wird über ein *Port-Demultiplexing* die gleichzeitige

Bedienung mehrerer Prozesse ermöglicht. Im Grunde besitzt UDP die Funktionalität einer Anwendungsschnittstelle zum IP, wobei folgende Anwendungen/Dienste unterstützt werden:

Dienst	Portnummer
Trivial File Transfer Protocol (TFTP)	69
Domain Name System (DNS)	53
Simple Network Management Protocol (SNMP)	161
Remote Procedure Calls (RPC)	111

Der Aufbau des UDP-Datagrammformats ergibt sich aus [Abb. 2–25](#).

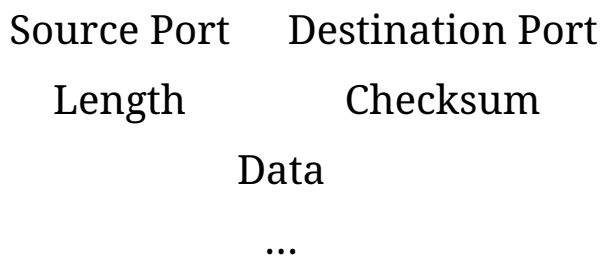


Abb. 2–25 Darstellung des UDP-Datagrammformats

- *Source Port* (16)
Portnummer des sendenden Prozesses
- *Destination Port* (16)
Portnummer des empfangenden Prozesses

- *Length* (16)
Datagrammlänge des Gesamt-UDP-Datagramms inkl. Header
- *Checksum* (16)
Checksumme der Daten, des UDP-Headers und des Pseudo-UDP-Headers

HINWEIS

Der Pseudo-Header besitzt eine Länge von 12 Bytes und besteht aus Fragmenten des IP-Headers. Er wird nicht über das Netzwerk übertragen.

Protokolle der Anwendungsschicht (Layer 5–7)

Spezielle Layer-5- oder Layer-6-Protokolle lassen sich nicht identifizieren, denn diese Protokollschichten (*Session* bzw. *Presentation*) stellen in der Regel Dienste zur Verfügung, die meist von der über- bzw. untergeordneten Protokollschicht verwendet werden. Man spricht beispielsweise von einer *TCP-Session* (bzw. -Connection) und bezeichnet damit ein Netzwerkobjekt, das weder der Schicht 4 noch der Schicht 5 eindeutig zugeordnet werden kann; es setzt sich vielmehr aus Diensten beider Schichten zusammen. Ähnlich verhält es sich mit dem Presentation Layer, der Darstellungsschicht. Als Beispiel mag hier das X-Protokoll dienen. Hier stellt die

grafische Oberfläche (*X-Windows*) als *Presentation Level* einen integralen Bestandteil des Protokolls dar.

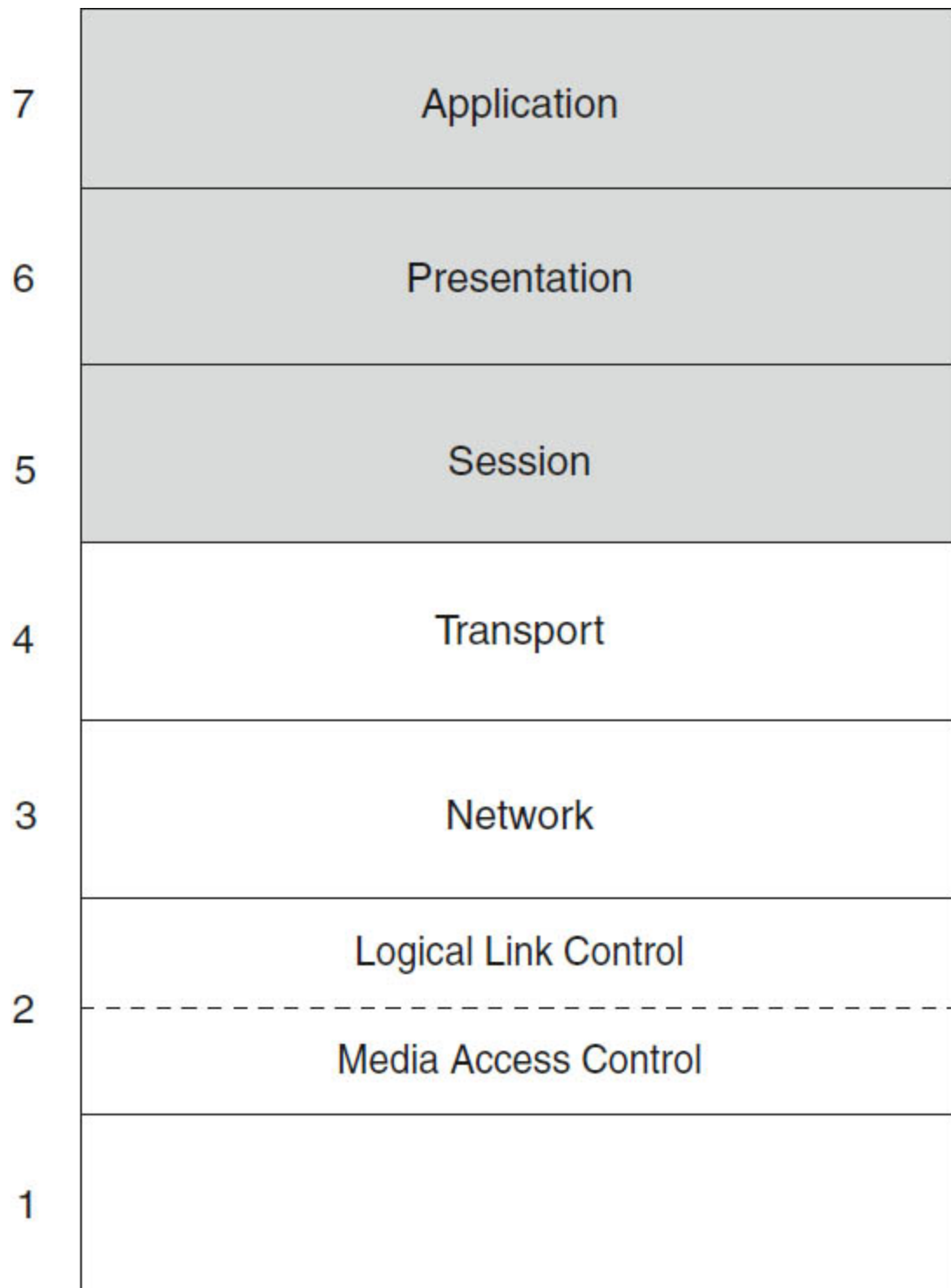


Abb. 2–26 *Layer 5 bis 7 (Session, Presentation, Application)*

Nachfolgend werden die wichtigsten Anwendungsprotokolle im TCP/IP-Umfeld erläutert, die ihrerseits verwendet werden können, um weitere komplexe Anwendungen zu entwickeln. Die Anwendungsschicht besitzt nach oben keine klar definierbare »Borderline« (zumindest nicht nach ISO/OSI), obwohl es Anwendungen gibt, die wiederum auf andere Anwendungen aufsetzen.

HINWEIS

Aufgrund des Umfangs, der Komplexität und der Bedeutung der verfügbaren Möglichkeiten der Protokolle und Anwendungen auf den höheren Layer-Schichten werden diese allesamt in einem eigenen Kapitel behandelt ([Kap. 6](#)).

Sonstige Protokolle

Abgesehen von den zuvor dargestellten Protokollen entlang des ISO/OSI-Schichtenmodells gibt es natürlich weitere Netzwerkprotokolle, die im Zusammenspiel heutiger Kommunikation eine große Rolle spielen. Sie sind oft nur sehr schwer einer Schicht innerhalb des ISO/OSI-Modells zuzuordnen (z.B. MPLS – *Multi-Protocol Label Switching*) oder repräsentieren kein reines Protokoll, sondern eher eine komplexe Protokoll-Architektur (z.B. SD-WAN – *Software*

Defined WAN). Einige heute wichtige Protokollstrukturen und -konzepte sind im Folgenden zusammengefasst dargestellt (siehe hierzu auch [Kap. 4](#) für weiterführende Details):

- MPLS – Das *Multi-Protocol Label Switching* hat sich neben einigen proprietären Lösungsansätzen unter Federführung des IETF (z.B. RFC 3031 vom Januar 2001) als eine zwischen Layer 2 und 3 eingefügte virtuelle Topologieunabhängige Kommunikationsschicht etabliert. Zahlreiche weiterführende RFCs haben bis heute die Spezifikationen, Charakteristika und Einsatzbereiche von MPLS weiterentwickelt. MPLS soll für eine maximale Flexibilität sorgen und dabei die Vermittlungsunabhängigkeit des Layer 3 und die hohe Performance des Layer 2 nutzen. Ihr Einsatzbereich liegt in Weitverkehrsnetzen (WAN = *Wide Area Networks*). Heute wird MPLS manchmal als veraltet angesehen und dem neueren SD-WAN-Konzept mehr Zukunftschancen eingeräumt. Allerdings gibt es derzeit kein vergleichbar leistungsfähiges WAN-Protokoll, das Qualitätsmerkmale berücksichtigen kann (*Quality of Service* – QoS); auch für Echtzeitanwendungen wie Voice oder Video.
- SDN – SD-LAN/SD-WAN – *Software Defined Networking* (SDN) wird als zukunftsorientierte Netzwerkarchitektur angesehen, die mit Fokus auf Zentrales Management, Automatisierung und einer Entkopplung von Software und Hardware die

Herausforderungen in Netzwerken der Zukunft bewältigen will. Dabei wird versucht, die Konzeption auf den LAN-Bereich (SD-LAN) und auf das WAN (SD-WAN) zu übertragen.

- SIP – Unter dem *Session Initiation Protocol* versteht man ein Protokoll der Anwendungsebene, das für den Aufbau, Betrieb und Abbau von Sitzungen (Signalisierung) mit einem oder mehreren Telefoniepartnern im Bereich der »Voice over IP«-Protokolle eingesetzt wird (siehe hierzu RFC 2543 aus 1999). Das Protokoll ist nicht ganz neu, besitzt aber heute deshalb große Relevanz, da die Sprachkommunikation sowie ihre Erweiterung in die Bereiche der Multimedia-Konferenzen (Skype, TEAMS, Zoom, WebEx usw.) an Bedeutung deutlich zugenommen hat. Dies ist nicht zuletzt der Tatsache geschuldet, dass in den letzten Jahren zunehmend Homeoffice-Arbeitsplätze eingerichtet wurden.
- DetNet – *Deterministic Networking* (siehe hierzu auch RFC 8655 aus 2019). DetNet wird auf ISO/OSI-Schicht 3 (Networking) eingesetzt und umfasst Protokolle auf den unteren Lower-Layer-Protokollen wie MPLS oder im IEEE-802.1-Standard (*Time-Sensitive Networking* – TSN). Es repräsentiert ein Regelwerk samt Protokollen für Echtzeit bzw. zeitkritische Kommunikation.
- NTP – Das *Network Time Protocol* ist bereits sehr früh (siehe dazu RFC 958 aus 1985 in einer ersten Version; die letzte

Version gem. RFC 5905 aus 2010 beschreibt die aktuelle NTP-Version 4) als Standard definiert worden und hat Einzug in zahlreiche Anwendungen gehalten. Zahlreiche Aktualisierungen haben das Protokoll im Laufe der Jahrzehnte reifen lassen. Es hat zuletzt eine aus Sicherheitsgründen erforderliche Erweiterung (*port randomization*) im RFC 9109 vom August 2021 erhalten.

Auf andere, ältere Protokolle wie X.25, Frame Relay oder in der Vergangenheit relevante Point-to-Point-Protokolle wird hier nicht näher eingegangen.

Adressierung im IP-Netzwerk

Der Begriff der Netzwerkadressierung oder der Adressvergabe wird an verschiedenen Stellen dieses Buches verwendet. Für das Verständnis eines allgemeinen Sachverhaltes im Zusammenhang mit adressierbaren Netzwerkkomponenten bedarf dieser Ausdruck keiner ausdrücklichen Erklärung. Um Näheres über die Adressierung im IP-Netzwerk zu erfahren, muss man sich detailliert mit dem Konzept und seinen Besonderheiten beschäftigen.

Adresskonzept

Die TCP/IP-Protokollfamilie kann grundsätzlich unter verschiedenen Netzwerktopologien eingesetzt werden, um damit eine wechselseitige Kommunikation der Teilnehmer eines Netzwerks zu gewährleisten. Allerdings gibt es eine Minimalforderung, die unbedingt erfüllt sein muss: *Die Adressierung der Knoten eines Netzwerks muss eindeutig sein.* So muss jede Netzwerkressource, sei es ein Rechner (PC), eine Workstation, ein Drucker oder auch ein Tablet, durch eine eindeutige Adresse bzw. durch einen eindeutigen Namen im Netzwerk identifizierbar sein. Wenn diese Grundvoraussetzung nicht gewährleistet ist, kann eine Kommunikation zwischen den Kommunikationspartnern nicht stattfinden.

Vergleichbar ist dieses Prinzip mit der Vergabe von Telefonnummern, bei der das Chaos perfekt wäre, wenn mehrere Rufnummern doppelt vergeben würden. Für Gesprächsteilnehmer in Orten mit unterschiedlichen Ortskennziffern (Vorwahlen) gilt dies natürlich nicht. Eine gültige Doppelbelegung ist dabei durchaus möglich, da die Eindeutigkeit durch eine stets eindeutige Ortskennziffer garantiert ist.

Adressierungsverfahren

Ganz wichtig ist, dass die physische von der logischen Adressierung im Netzwerk unterschieden werden muss. Unter einer *physischen Adressierung* versteht man die mit einer Netzwerkressource unverwechselbar verbundene eindeutige Kennung, meist sogar weltweit, die zur Identifikation in einem Netzwerk herangezogen wird. Bei den heute handelsüblichen Netzwerkcontrollern ist eine solche Hardwareadresse (MAC-Adresse) fest in einer Komponente hinterlegt, die auch nicht modifiziert werden kann. So werden beispielsweise für Rechner Netzwerkadapter (Einschubkarten) mit *burnt-in addresses* (eingebrannten Adressen) vertrieben, die zwecks Anschlusses an ein Netzwerk in die entsprechenden Rechner eingebaut werden können. Das Gleiche gilt für andere Geräte, die teilweise bereits mit fest eingebauten Netzwerkcontrollern

ausgestattet sind, wie beispielsweise Smartphones, Tablets oder mittlerweile auch Digitalkameras.

Im Gegensatz dazu ist die *logische Adressierung* in erster Linie vom Netzwerkprotokoll und seinen charakteristischen Eigenschaften abhängig. Sie hat zunächst einmal nichts mit der physischen Adresse zu tun, und es liegt allein im Verantwortungsbereich des Netzbetreibers, für die notwendige Eindeutigkeit der Adressen zu sorgen. Der Systemverwalter eines Netzwerks ist damit in der Lage, eine für ihn bzw. für das Unternehmen sinnvolle Adressstruktur zu entwickeln und dann die jeweils vorgesehene logische Adresse der physischen Adresse »überzustülpen«.

In TCP/IP-Netzwerken kommt zur logischen Adressierung die *IP-Adresse* zum Einsatz. In der IP-Version 4 besteht diese Adresse aus vier Oktetten, die in dezimaler Form (*Dotted Notation*) formuliert wird (Beispiel: 192.205.76.6). Die Weiterentwicklung in diesem Bereich hin zur IP-Version 6 nutzt einen erweiterten Adressraum von 16 Oktetten.

Ein oft vernachlässigter Aspekt, insbesondere in der Phase der Netzwerkplanung, ist das System der *Host-Namen* und der *Domänen*. Dabei wird der logischen IP-Adresse zusätzlich ein sogenannter *fully qualified host name* (vollqualifizierter Name)

zugeordnet, der aus einer durch Punkt separierten Folge von Ordnungsnamen besteht. Diese Namen setzen sich in der Regel aus einer DNS-Domäne (z.B. *firma.de*) und einem Host-Namen (z.B. *host1*) zusammen. Die Adressierung auf diesem Level kann sogar den einzelnen Anwender auf dem System identifizieren: *anwender@host1.firma.de*. In der Praxis wird dieser Adresstyp meist aus Anwendungen heraus zur Adressierung verwendet.

HINWEIS

Nähere Angaben zu DNS (*Domain Name System*) und der Zuordnung von Domänen enthält *Kapitel 5*.

Zusammenfassend lassen sich für die Adressierung in einem TCP/IP-Netzwerk folgende Layer-abhängige Verfahren nennen:

- physische Adressierung (*Lower Data Link Layer*)
- logische IP-Adressierung (*Network Layer*)
- logische Adressierung über Host Names und Domains (*Network Layer*)

In der hier genannten Reihenfolge bedeuten die einzelnen Adressierungsverfahren eine steigende Flexibilität gegenüber Modifikationen im Netzwerk. Der (möglicherweise relativ häufige) Austausch von Netzwerkcontrollern durch Defekt oder Wechsel der Hardware lässt die zugeordnete IP-Adresse davon

unberührt. Sie bleibt bestehen, auch wenn sich die Hardware ändert. Werden allerdings Erweiterungen oder Modifikationen in der IP-Netzstruktur notwendig, so ist es möglich, dass dies unter Umständen auch eine Änderung der IP-Adresse nach sich zieht. Der symbolische Name der IP-Adresse oder gar des Anwenders ist davon natürlich nicht betroffen.

Adressregistrierung

Der Ursprung des Internet Protocol (IP) lag in einem vom amerikanischen Verteidigungsministerium initiierten Netzwerk und bedurfte einer besonderen Kontrollinstanz. Diese Instanz stellte zunächst das *Internet Architecture Board* (IAB) mit seinen zahlreichen Unterabteilungen und Nebeninstitutionen dar, die für eine reibungslose und sichere Funktion des Internetnetzwerks verantwortlich war. Nach und nach übernahm das ICANN (*Internet Corporation for Assigned Names and Numbers*) diese Aufgabe, zunächst noch unter Einfluss der US-Regierung, seit 2016 jedoch davon unabhängig. Zu den Aufgaben gehört u.a. die Gewährleistung der Eindeutigkeit von IP-Adressen im öffentlichen Datenverkehr, also in der Anbindung bzw. Integration von Firmennetzwerken in öffentliche Netzwerke. Weitere Details hinsichtlich Aufgaben, Struktur und Mitglieder der ICANN lassen sich unter ihrer Homepage im Internet nachlesen (www.icann.org).

Zu Beginn einer Überlegung zum Aufbau eines TCP/IP-Netzwerks sollte grundsätzlich die Frage stehen: Soll das geplante Netzwerk öffentlich registriert werden oder ist es lediglich als ein unternehmensinternes IP-Netzwerk auszulegen? In der Praxis zeigt sich immer wieder, dass es durchaus sinnvoll sein kann, bereits in der Anfangsphase der Netzwerkplanung an eine offizielle Registrierung von Domänen und Subnetzen zu denken und die erforderlichen Planungen durchzuführen.

Eine Alternative zur völligen Netzöffnung ist der Einsatz eines IP-Gateways, das für eine Adressumsetzung der internen Adressen in registrierte Adressen verantwortlich ist (NAT = *Network Address Translation*).

HINWEIS

Obwohl die Internet-Registrierung in Eigenregie erfolgen kann, ist es jedoch zu empfehlen, den Gesamtprozess inklusive der Netzwerkplanung in Zusammenarbeit mit einem erfahrenen Service-Provider abzuwickeln.

Adressaufbau und Adressklassen

Eine IP-Adresse der IP-Version 4 besteht aus einer 32-Bit-Sequenz, die in vier Gruppen zu je acht Bits aufgeteilt ist. Jede

dieser Gruppen wird als *Oktett* bezeichnet; ihre Darstellungsweise ist dezimal. Die logische Gliederung dieser Adresse erfolgt in eine *Netz-ID* und eine *Host-ID*, wobei der Umfang ihres Adressbereichs aus fünf Adressklassen hervorgeht:

- *Klasse A*

Klasse A charakterisiert einen Adresstyp, der einen sehr kleinen Adressbereich für die Netz-ID (7 Bits) und einen großen Adressbereich für die Host-ID besitzt. Die Anzahl definierbarer IP-Hosts (IP-Adressen) ist somit extrem hoch. Die Klassifizierung geht eindeutig aus den ersten Bits des ersten Oktetts der Adresse hervor. In dieser Klasse A ist das erste Bit reserviert (besitzt den Wert »0«), sodass lediglich die restlichen sieben Bits für eine Netz-ID zur Verfügung stehen (maximaler Wert also binär »1111111« oder dezimal 128). Daraus ergibt sich eine Anzahl von theoretisch 128 Netzen, von denen jedoch die 0 und 127 wegfallen, sodass letztlich 126 Netzwerke adressiert werden können (siehe [Abb. 3–1](#)).

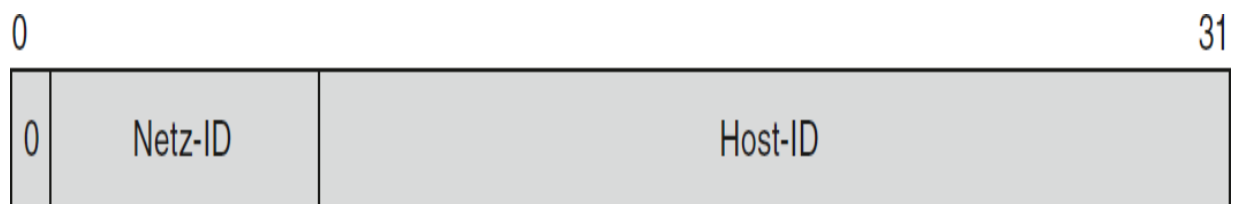


Abb. 3–1 Schema einer IP-Adresse (Version 4) gemäß Klasse A

HINWEIS

Die Klasse A steht für Neuzulassungen praktisch nicht mehr zur Verfügung, da sie bereits vollständig durch Registrierungen von Unternehmen der »ersten Stunde« belegt ist.

- *Klasse B*

In dieser Klasse sind die ersten beiden Bits des ersten Oktetts reserviert. Die »Klassen-Bits« besitzen den Wert »10«. Somit lässt sich maximal der Wert »10111111« für das erste Oktett erzeugen; dies entspricht dezimal der Zahl 191. Aufgrund der Erweiterung der Netz-ID um ein weiteres Oktett können nunmehr alle Netze im Bereich zwischen 128.1 und 191.255 adressiert werden. Dies entspricht einer Anzahl von 16.384 Netzwerken gegenüber 126 bei Klasse A. Die Anzahl von definierbaren IP-Hosts pro Netzwerk liegt in dieser Klasse B bei 65.534 (Beispiel: Die Netz-ID 154.8 umfasst einen Adressbereich von 154.8.0.1 bis 154.8.255.254). Diese Klasse eignet sich für mittelgroße Netze mit einer mittleren Anzahl von IP-Hosts (siehe [Abb. 3-2](#)).

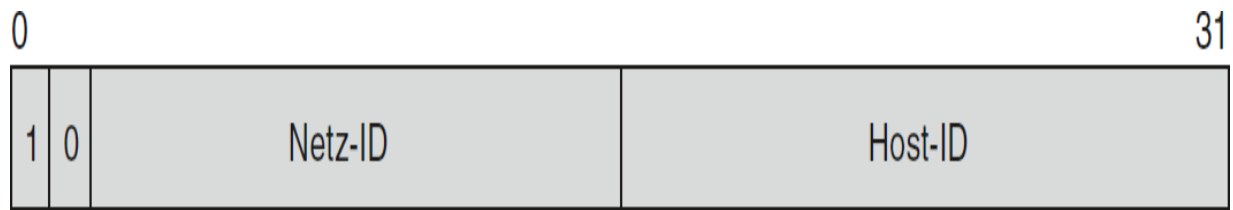


Abb. 3–2 Schema einer IP-Adresse (Version 4) gemäß Klasse B

HINWEIS

Genau wie bei Klasse A stehen auch Klasse-B-Adressen für Neuregistrierungen praktisch nicht mehr zur Verfügung.

- *Klasse C*

Bei *Klasse C* erfolgt eine Reservierung der ersten drei Bits des ersten Oktetts auf die Werte »110«, so wie in [Abbildung 3–3](#) dargestellt. Der maximal erreichbare Wert lautet demnach binär »11011111« und dezimal 223. Für die um ein weiteres Oktett erweiterte Netz-ID ergibt sich ein Anfangswert von 192.0.1 und eine theoretische Obergrenze von 223.255.255. Maximal können daher 2.097.152 Netze abgebildet werden, allerdings lediglich mit jeweils 254 Hosts. Aus diesem Kontingent wurden bzw. werden die meisten Registrierungsanträge befriedigt.

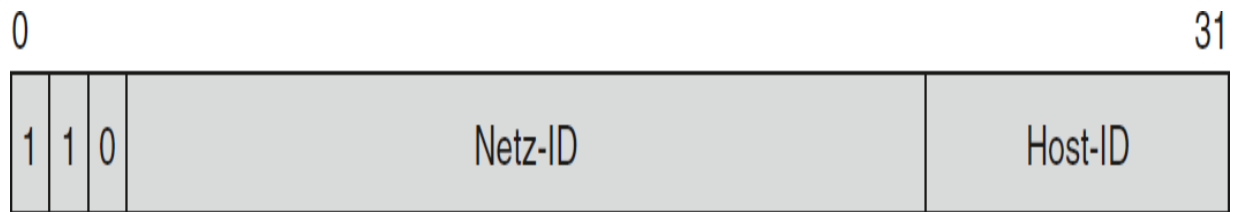


Abb. 3–3 Schema einer IP-Adresse (Version 4) gemäß Klasse C

- *Klasse D*

Bei Klasse D sind die ersten vier Bits des ersten Oktetts auf binär »1110« reserviert (siehe [Abb. 3–4](#)). Hier existiert allerdings kein Definitionsbereich für eine Netz-ID. Die hier generierbaren IP-Adressen zwischen 224.0.0.0 und 239.255.255.254 besitzen Sonderstatus, werden normalerweise als »Multicast-Adressen« bezeichnet (siehe auch [Kap. 4 Routing](#)) und stehen für besondere Funktionen zur Verfügung.



Abb. 3–4 Schema einer IP-Adresse (Version 4) gemäß Klasse D

- *Klasse E*

Auch die Klasse E genießt Sonderstatus und steht für den Allgemeingebrauch nicht zur Verfügung. Die Reservierung erstreckt sich auch hier auf die ersten vier Bits des ersten Oktetts, allerdings erhält das vierte Bit den Wert »1«, sodass

die erste benutzbare Adresse 240.0.0.0 und die letzte Adresse mit 255.255.255.254 angegeben werden kann.

In der nachfolgenden Tabelle sind die verschiedenen Klasseneinteilungen und die damit verbundene Adresszuordnung zusammengefasst:

Klasse	Klassen-Bits	Anzahl Netze	Anzahl Hosts pro Netz
A	0	126	16.777.214 (2563-2)
B	10	16.384 (64 × 2561)	65.534 (2562-2)
C	110	2.097.152 (32 × 2562)	254 (2561-2)
D	1110	-	-
E	1111	-	-

Subnetzadressierung

Die Klassifizierung der IP-Adressen gemäß dem im letzten Abschnitt beschriebenen System in Klasse-A-, Klasse-B- und Klasse-C-Adressen (die Sonderklassen D und E bleiben unberücksichtigt, da sie nicht praktisch nutzbar sind) ist relativ starr und unflexibel. Betrachtet man die Entwicklung der öffentlichen Adressvergabe und Registrierung innerhalb der letzten Jahrzehnte, so ist es mittlerweile nahezu unmöglich,

eine der überaus attraktiven (öffentlichen) Klasse-A-Adressen zu erhalten; selbst B-Adressen sind nur noch sehr schwer zu bekommen.

Adressen der Klasse C besitzen grundsätzlich einen stark eingeschränkten Adressraum (maximal 254 adressierbare Hosts pro Adresse), sodass man bei der Einrichtung umfangreicher IP-Netzwerke leicht an seine Grenzen stößt. Darüber hinaus scheint sich das Routing innerhalb des öffentlichen Internets zu einem Problem zu verdichten, da der Umfang der Adressinformationen (*Routing-Tabellen*) ein derartiges Ausmaß angenommen hat, dass er von der gegenwärtig verfügbaren Soft- und Hardware nur noch schwer zu verwalten ist. Es ist ferner zu erwarten, dass der zurzeit verfügbare Adressraum von 32 Bits innerhalb der nächsten Zeit sicher nicht mehr ausreichen wird, um den Bedarf an registrierten IP-Adressen zu befriedigen.

HINWEIS

Die Probleme bei der Adressvergabe unter IP, Version 4, und mögliche Lösungsansätze sind unter anderem im RFC 1519 vom September 1993 beschrieben: *Classless Inter-Domain*

Routing (CIDR): An Address Assignment and Aggregation Strategy.

Eine mögliche Lösung der Adressproblematik von IP, Version 4, liegt in der Bildung von Subnetzen, die das statische Klassenkonzept durchbrechen. Danach ist es möglich, innerhalb einer verfügbaren Klassenadresse alter Konvention weitere Subnetze zu definieren und die Anzahl adressierbarer Hosts auf die Subnetze aufzuteilen. Dazu passt der RFC 1817 vom August 1995, der die CIDR-Betrachtungen um die Fähigkeiten relevanter Routing-Protokolle erweitert, indem festgehalten wird, dass Protokolle wie RIP, BGP-3, EGP und IGRP nicht für das CIDR-Konzept geeignet sind. Neuere Protokolle, wie OSPF, RIP II, Integrated IS-IS und Enhanced IGRP, lassen sich allerdings verwenden.

HINWEIS

Weitergehende Informationen zum Thema *Routing* und den Funktionen und Möglichkeiten der verschiedenen Routing-Protokolle enthält [Kapitel 4](#).

Der RFC 4632 vom August 2006 liefert wichtige Informationen zum praktischen Einsatz des CIDR-Konzepts (siehe auch hierzu Aktivitäten der ROAD Workgroup in [Abschnitt 3.4.2.2](#)).

Prinzip

Angenommen, ein Unternehmen mittlerer Größe hat zwecks Aufbau eines IP-Netzwerks beim DENIC (*Deutsches Network Information Center*) eine öffentliche Adresse beantragt. Folgende Klasse-B-Adresse wird zugewiesen: 190.136.0.0. Mit dieser Adresse lassen sich bekanntlich 65.536 Hosts adressieren – eine Zahl, die nicht unbedingt dem Bedarfsprofil eines größeren Unternehmens entspricht, das durch den Einsatz von Routern das eigene Netzwerk strukturieren möchte. Für jeden Router muss ein expliziter Netzübergang definiert werden, d.h., der Router umfasst, sofern er lediglich zwei Netzwerke miteinander verbindet, zwei IP-Adressen mit zwei unterschiedlichen Netz-IDs. Da in diesem Beispiel lediglich ein einziges logisches Klasse-B-Netzwerk verfügbar ist, muss das gewünschte Ziel dadurch erreicht werden, dass überall dort, wo Router eingesetzt werden sollen, Subnetze gebildet werden.

Typen und Design der Subnetzmaske

Das zur Bildung von Subnetzen erforderliche Instrumentarium ist die *Subnetwork Mask* (Subnetzmaske). Sie stellt, ebenso wie die IP-Adresse, eine Folge von 32 Bits dar, die in der Regel gemeinsam mit der IP-Adresse auf den einzelnen Systemen (Router, Hosts usw.) konfiguriert wird. Ihre Schreibweise ist der

IP-Adresse angepasst: *dotted decimal*. Die Funktionsweise einer Subnetzmaske ist allerdings wesentlich besser nachvollziehbar, wenn man ihren Binärcode betrachtet. Die zu vier Oktetten zusammengefasste »Maske« codiert nämlich überall dort, wo die IP-Adresse als Netz-ID interpretiert werden soll, eine binäre 1. Der restliche Teil der *Subnetwork Mask* bleibt für die Host-ID übrig, also genau die Stellen, wo binäre Nullen codiert sind.

Für den Fall, dass lediglich mit der registrierten IP-Adresse, also ohne »Subnetting«, gearbeitet wird, gilt eine *Default Subnetwork Mask*, denn der IP-Prozess eines Systems berücksichtigt auch dann eine Subnetzmaske, wenn keine eigene Maske definiert wurde. Sie entspricht dann der jeweiligen Klasse, so wie nachfolgend dargestellt:

Klasse	dotted decimal	binary
A	255.0.0.0	11111111 00000000 00000000 00000000
B	255.255.0.0	11111111 11111111 00000000 00000000
C	255.255.255.0	11111111 11111111 11111111 00000000

Soll jedoch eine individuelle Subnetzmaske verwendet werden, so erfolgt die Verteilung der 1er-Maske nicht mehr Byte-,

sondern Bit-orientiert:

B 255.255.192.0 **11111111 11111111 11000000**
 00000000

Zurück zum Beispiel: Die bei der Klasse-B-Adresse theoretisch verfügbaren 65.536 Hosts sollen auf verschiedene Subnetze aufgeteilt werden. Aufgrund von Überlegungen zur Infrastruktur des gewünschten Netzwerks wird die Entscheidung getroffen, mindestens fünf Subnetze zu bilden (siehe [Abb. 3–5](#)).

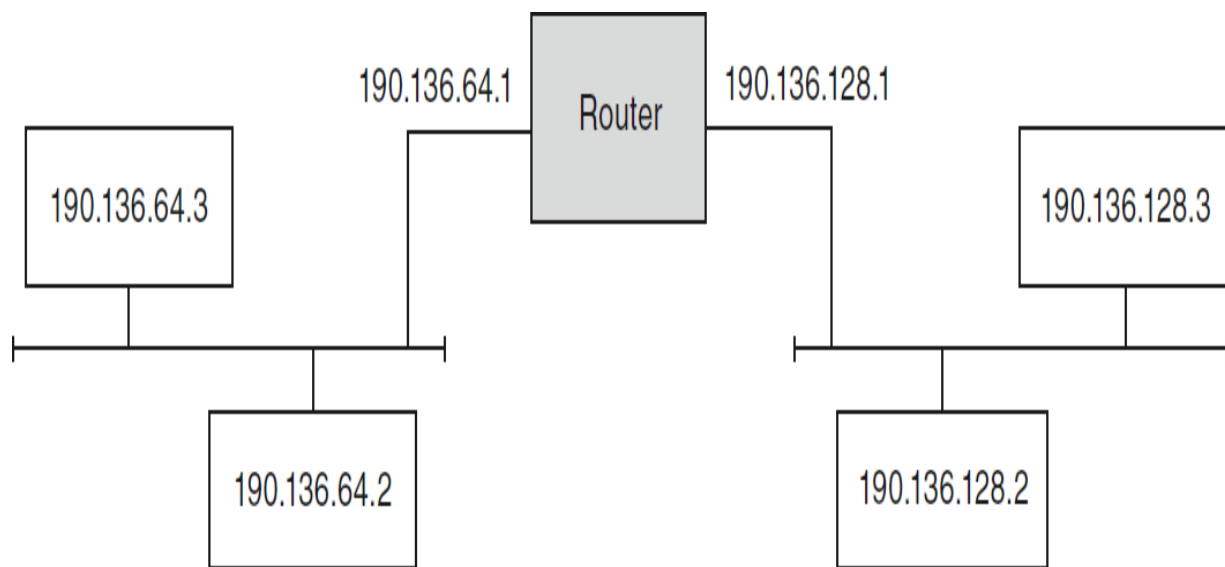


Abb. 3–5 *Aufbau und Struktur des gewünschten Netzwerks*

In diesem Fall bedeutet dies, dass vom dritten Byte der Subnetzmaske die ersten beiden Bits für die Netz-ID verwendet werden sollen; daraus ergibt sich eine Anzahl von maximal vier

Subnetzen. Die restlichen sechs Bits des dritten Bytes und das vollständige vierte Byte führen zu einer Anzahl von maximal 16.384 adressierbaren Hosts pro Subnetz (64 x 256; 64 aus dem dritten und 256 aus dem vierten Oktett). Die geplante Subnetwork Mask ergibt daher folgende Teilnetze:

Netz-ID	Host-ID
190.136.0.0	10111110 10001000 00 000000 00000000
190.136.64.0	10111110 10001000 01 000000 00000000
190.136.128.0	10111110 10001000 10 000000 00000000
190.136.192.0	10111110 10001000 11 000000 00000000

Da die Mindestanzahl von fünf definierbaren Subnetzen nicht erreicht werden kann, muss der Plan, eine Subnetzbildung mit den ersten beiden Bits des dritten Bytes der Subnetzmaske durchzuführen, verworfen werden. Als Ausweg bleibt die Hinzunahme eines weiteren Bits in der Subnetzmaske (im dritten Byte), um mindestens fünf Subnetze erzeugen zu können. Daraus wiederum ergibt sich die folgende Aufteilung:

Netz-ID	Host-ID
Subnetwork Mask:	11111111 11111111 111 00000 00000000
Subnetz	
190.136.0.0	10111110 10001000 000 00000 00000000
190.136.32.0	10111110 10001000 001 00000 00000000

190.136.64.0	10111110 10001000 01 000000 00000000
190.136.96.0	10111110 10001000 011 00000 00000000
190.136.128.0	10111110 10001000 100 00000 00000000
190.136.160.0	10111110 10001000 101 00000 00000000
190.136.192.0	10111110 10001000 110 00000 00000000
190.136.224.0	10111110 10001000 111 00000 00000000

Aus dieser Aufstellung geht hervor, dass die angestrebten fünf Subnetze gebildet werden können (maximal acht Subnetze). In jedem Subnetz müssen zwei Adressen gestrichen werden, wobei es sich dabei theoretisch jeweils um die erste (z.B. 190.136.96.0) und die letzte Adresse eines Subnetzes (z.B. 190.136.127.255) handelt. Dies ist notwendig, da der Wert »0« das jeweilige Subnetz bezeichnet und »255« die lokale Broadcast-Adresse repräsentiert, also beide für eine Adressierung nicht zur Verfügung stehen.

Unter diesen Voraussetzungen können als Ergebnis für die weitere Planung der Subnetzstruktur folgende Bereiche verwendet werden:

Nr.	Netz-ID	erster Host	letzter Host	Local Broadcast
1	190.136.0.0	190.136.0.1	190.136.31.254	190.136.31.255
2	190.136.32.0	190.136.32.1	190.136.63.254	190.136.63.255
3	190.136.64.0	190.136.64.1	190.136.95.254	190.136.95.255
4	190.136.96.0	190.136.96.1	190.136.127.254	190.136.127.255
5	190.136.128.0	190.136.128.1	190.136.159.254	190.136.159.255
6	190.136.160.0	190.136.160.1	190.136.191.254	190.136.191.255
7	190.136.192.0	190.136.192.1	190.136.223.254	190.136.223.255
8	190.136.224.0	190.136.224.1	190.136.255.254	190.136.255.255

Jedes der acht Subnetze kann somit 8.190 Hosts adressieren (8.192, um die Netz-ID und den lokalen Broadcast reduziert). Die Wahl einer anderen Subnetzmaske könnte eine völlig andere Aufteilung ergeben. Wird beispielsweise nicht die Subnetzmaske 255.255.192.0, sondern 255.255.255.240 verwendet, ergeben sich folgende Subnetze:

Netz-ID	Host-ID
Subnetwork Mask:	11111111 11111111 11111111 11110000
Subnetz	
190.136.0.0	10111110 10001000 00000000 00000000

190.136.0.16	10111110 10001000 00000000 0001 0000
190.136.0.32	10111110 10001000 00000000 001 00000
190.136.0.48	10111110 10001000 00000000 0011 0000
190.136.0.64	10111110 10001000 00000000 01 000000
...	
...	
190.136.0.224	10111110 10001000 00000000 111 00000
190.136.0.240	10111110 10001000 00000000 1111 0000

Analog erfolgt die Subnetzbildung für die weiteren Werte 2 bis 255 im dritten Oktett. Es können also bei einem vorliegenden Klasse-B-Netzwerk 190.136.0.0 durch *Subnetting* mit der Mask 255.255.255.240 insgesamt 4.096 Subnetze (256×16 ; 256 aus dem dritten und 16 aus dem vierten Oktett) mit jeweils 16 minus 2, also 14 IP-Adressen gebildet werden.

Verwendung privater IP-Adressen

Neben den öffentlich vergebenen und genutzten IP-Adressbereichen gibt es eine Vielzahl privater Subnetze, die ausschließlich zur internen Verwendung in Unternehmensnetzwerken genutzt werden können. Diese werden im Internet nicht berücksichtigt und können daher auch mehrfach vergeben werden, da sie ja die Unternehmensnetze nicht verlassen. Bei diesen reservierten

privaten Adressbereichen handelt es sich um folgende Subnetze:

- Klasse A: 10.0.0.0/255.0.0.0
- Klasse B: 172.16.0.0/255.240.0.0
- Klasse C: 192.168.0.0/255.255.0.0

Beim Verlassen des Unternehmensnetzwerks bzw. bei der Anbindung an das öffentliche Netzwerk (Internet) erfolgt in der Regel ein »Übersetzen« der privaten Adressen auf öffentliche, legal zugewiesene Adressen. Ein Verfahren, das in diesem Zusammenhang eingesetzt wird, ist das Prinzip der *Network Address Translation* (NAT), also einer Art »Übersetzungstabelle«.

Werden beispielsweise zwei Standorte eines Unternehmens über IP-Router verbunden, wobei die beiden Standorte über ein eigenes Netzwerk verfügen, die wiederum mit einem Standort-Backbone verbunden sind, so erfolgt über diesen Backbone die WAN-Verbindung über einen dedizierten Router. Für das interne IP-Netzwerk wird die Klasse-A-Adresse 10.0.0.0 verwendet. Für die Strukturierung in verschiedene Subnetze wird die Subnetzmaske 255.255.255.0 verwendet. Pro Abteilung (drittes Oktett) innerhalb eines Standorts (zweites Oktett) stehen somit maximal 254 Host-IDs (viertes Oktett) zur

Verfügung. Für den jeweiligen Standort lassen sich 255 verschiedene Abteilungsnetze etablieren. Für den Expansionsdrang des Unternehmens ist ebenfalls gesorgt: 255 Standorte können für die Zukunft mit einer eigenen Netz-ID adressiert werden. Es ergibt sich also folgende Adressstruktur:

Standort	Abteilung	Netz-ID
Hamburg	Auftragsbearbeitung	10.2.1.0
Hamburg	Buchhaltung	10.2.2.0
Hamburg	Versand	10.2.3.0
Hamburg	Datenverarbeitung	10.2.4.0
München	Auftragsbearbeitung	10.3.1.0
München	Buchhaltung	10.3.2.0
München	Versand	10.3.3.0
München	Datenverarbeitung	10.3.4.0
WAN-Verbindung der Standorte		10.1.1.0

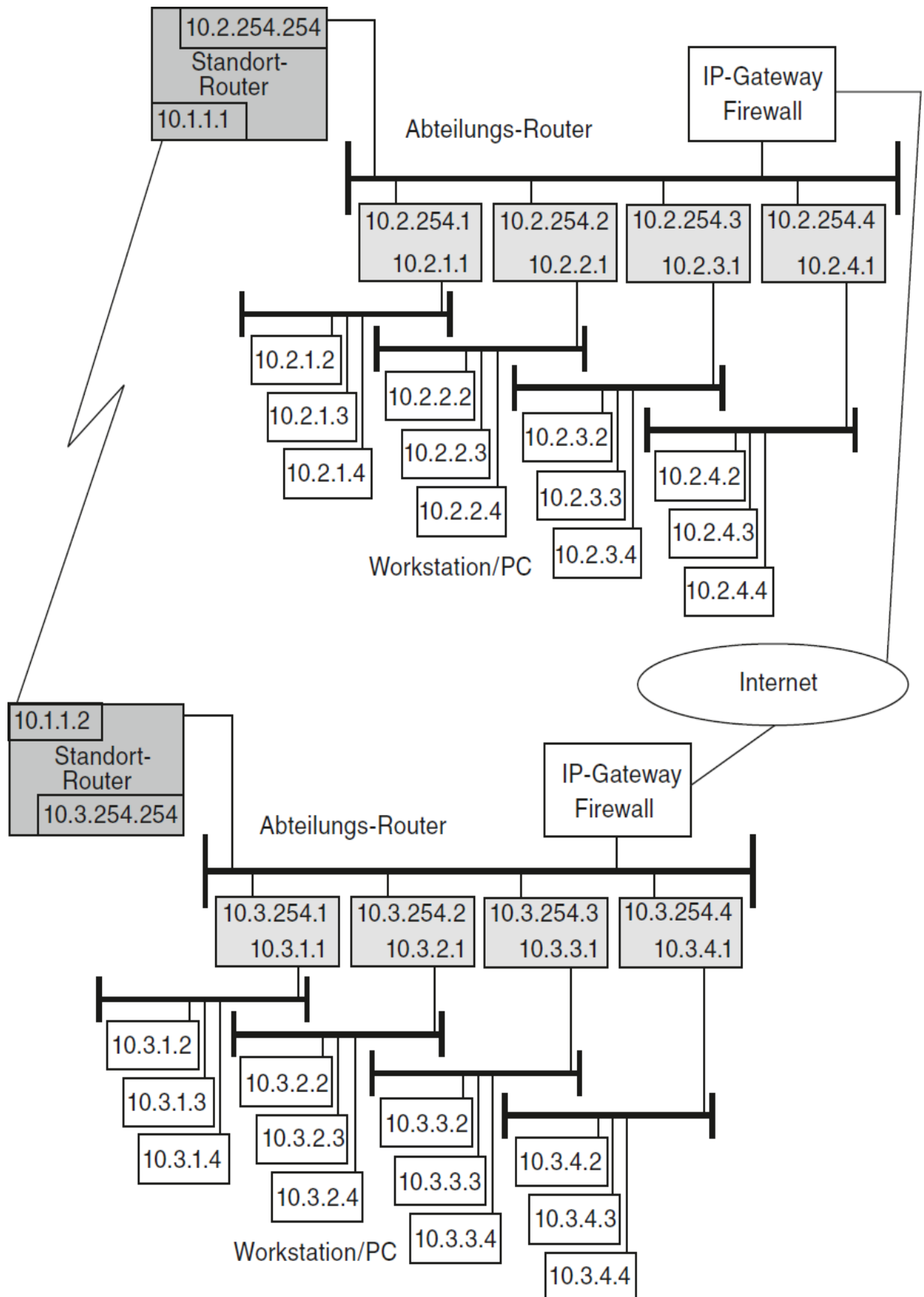


Abb. 3–6 *Beispiel: Adressstruktur bei einer WAN-Anbindung zweier Standorte*

Internetdomain und Subnetz

Ein erster Schritt zur eigenen Internetdomain ist zumeist die Kontaktaufnahme mit einem entsprechenden *Provider* bzw. *Hoster*, der in der Regel alle erforderlichen Formalitäten übernimmt. Somit entfällt ein direkter Vertragsabschluss mit dem DENIC, dem *Deutschen Network Information Center*. Das DENIC ist innerhalb Deutschlands für die kontrollierte Vergabe von Subdomains unterhalb der Top Level Domain »de« zuständig.

Will man eigene Internetdienste betreiben bzw. eigene Server ins Internet bringen (z.B. einen FTP- oder Webserver), so werden dazu legale Internetadressen benötigt. Die Vergabe privater Adressen (gemäß RFC 1918) ist zu diesem Zweck nicht möglich, da man schließlich für weltweite Eindeutigkeit dieser IP-Adressen sorgen muss. Diese kann aber nur dann gewährleistet werden, wenn eine zentrale Instanz darüber wacht, dass eine IP-Adresse stets nur einmal vergeben wird. Für diese Aufgabe ist die ICANN verantwortlich (siehe auch hierzu [Abschnitt 3.1.2](#)).

Diese »Non-Profit-Organisation« verwaltet weltweit zentral eine Datensammlung von IP-Adressen und Subnetzen, wobei sie die Vergabe in anderen Ländern an weitere Instanzen delegiert. Dabei handelt es sich um das APNIC (*Asia-Pacific Network Information Center*; www.apnic.net), das ARIN (*American Registry for Internet Numbers*; www.arin.net), LACNIC (*Latin America and Caribbean Network Information Centre*; www.lacnic.net), AfriNIC (*African Network Information Centre*; www.afrinic.net) und das RIPE NCC (*Réseaux IP Européens*; www.ripe.net). Letzteres übernimmt diese Aufgabe für den europäischen Bereich des Internets.

Die wichtigsten Informationen zur IP-Adressierung, -Organisation und zu ihren Besonderheiten sind in den RFCs 2050 (*Internet Registry IP Allocation Guidelines*), 1918 (*Address Allocation for Private Internets*) und 1518 (*An Architecture for IP Address Allocation with CIDR*) hinterlegt. Es existieren auch weitere RFCs zu verschiedenen Registrierungsaufgaben. Diese kann man am besten im öffentlich verfügbaren RFC-Index nachlesen.

Da mit der Domainbeschaffung für Privatpersonen normalerweise kein eigenes IP-Subnetz (mehrere öffentliche IP-Adressen, die ausschließlich für den Antragsteller reserviert sind) erworben wird und der Kunde für den Internetzugang

(nicht für die Bereitstellung eigener Dienste) dynamische IP-Adressen aus dem eigenen Kontingent des Providers erhält, wird der Prozess der Domain-Beantragung und Reservierung vom Provider meist in Eigenregie durchgeführt.

Dynamische Adressvergabe

Neben der statischen Zuordnung von Adressen für den Bereich der TCP/IP-Protokollfamilie gibt es je nach Anforderung auch andere Formen der Adresszuordnung, wie beispielsweise eine dynamische Vergabe der Adressen. Die beiden grundlegenden oder bekanntesten Verfahren sind dabei BootP und DHCP, wobei sich die Zuordnung nicht nur auf eine IP-Adresse bezieht, sondern damit auch weitere Angaben zur Netzwerkkonfiguration übermittelt werden können.

HINWEIS

Innerhalb von IP-Netzwerken folgt die Adressvergabe grundsätzlich sehr stringenten Vorgaben, die beispielsweise festlegen, dass in einem Netzwerk (IP-Segment) keine IP-Adresse doppelt vergeben werden darf. Sind IP-Adressen doppelt vorhanden, führt dies in der Regel zu den merkwürdigsten Effekten bis hin zum Ausfall der betreffenden Endgeräte.

Bootstrap Protocol (BootP)

Das *Bootstrap Protocol* (BootP) wurde als UDP-basierendes Protokoll ausschließlich entwickelt, um Startvorgänge (Booten) zu realisieren, die dazu benötigt werden, sogenannte *Diskless Workstations* (Rechner ohne Disketten- und Festplattenlaufwerke) als Arbeitsstationen im Netzwerk zu betreiben. Der Schwerpunkt lag bzw. liegt dabei auf dem Einsatz in UNIX-Umgebungen.

Mit BootP wird nicht der eigentliche Netzwerkstart (Booten) durchgeführt, sondern lediglich die Übernahme relevanter Netzwerkkonfigurationsdaten (z.B. eigene IP-Adresse oder IP-Adresse des Boot-Servers). Der Ladevorgang selbst wird mit klassischen Transferprotokollen durchgeführt, bei denen es sich meistens um TFTP handelt.

HINWEIS

Nähere Informationen zu TFTP (*Trivial File Transfer Protocol*) enthält [Kapitel 6](#).

Der BootP-Client beginnt seinen Boot-Vorgang mit einem IP-Broadcast, also einer Abfrage an alle Geräte im Netzwerk. Dabei gibt er die Hardwareadresse (MAC-Adresse) des eigenen Netzwerkcontrollers an. Derjenige Boot-Server, der das Gerät

erkennt, antwortet mit einem *Reply* und schickt dem Client die für ihn vorgesehene IP-Adresse, den Namen (bzw. die IP-Adresse) des Boot-Servers und der Boot-Datei, die geladen werden soll. Nach Abschluss des nachfolgenden Ladevorgangs ist die *Diskless Workstation* ein vollwertiger IP-Knoten und kann an der Netzwerkkommunikation teilnehmen.

Die Aktivitäten des BootP-Clients (RFC 951) sind am einfachsten folgendermaßen zu charakterisieren:

- *Name*

Für die letztlich physische Adressierung in einem Netzwerk ist die sog. Hardwareadresse verantwortlich, die zumeist im ROM des Netzwerkcontrollers eingebrannt ist (*burnt-in address*). Diese Adresse muss der Rechner lesen können, bevor er sich ins Netzwerk begibt, wobei der Leseprozess durch Aktivierung eines separaten BootP-Vorgangs vorgenommen wird.

- *Netzwerkfähigkeit*

Über einen *BootP Request* wird netzwerkweit die Frage gestellt: *Wer kennt mich?* Der eigene Name (*burnt-in*) ist Bestandteil der Anfrage. Ist der Name einem anderen Rechner bekannt (per Definition), so erfolgt ein *Reply* an den anfragenden Rechner. Dieser enthält die eigene IP-Adresse, die IP-Adresse des BootP-Servers, ggf. die IP-Adresse eines

Routers und den Namen der Datei, die gebootet werden soll. In einigen Maschinen lassen sich diese Informationen auch manuell vorkonfigurieren. Mit dieser Basiskonfiguration lassen sich nun auch *ARP-Requests/-Replies* bearbeiten und TFTP kann benutzt werden.

- *Weitere Angaben*

Nachdem alle notwendigen Informationen für den Boot-Vorgang vorliegen, kann dieser über TFTP vorgenommen werden. Während des Ladeprozesses erfolgt die Übernahme weiterer wichtiger Netzwerkinformationen. Der eigene Rechner wird initialisiert. Im nächsten Schritt wird durch die Etablierung von Netzwerkverbindungen (z.B. TELNET über TCP) und Sitzungen, die eine grafische Oberfläche präsentieren (z.B. X11/Motif), eine funktionsfähige Arbeitsumgebung hergestellt. In [Abbildung 3-7](#) ist die *BootP Message* dargestellt.

Operation	Htype	Hlen	Hops
Transaction ID			
Seconds		– unused –	
Client IP-Address			
Your IP-Address			
Server IP-Address			
Gateway IP-Address			
Client Hardware Address (4 x 32 Bit)			
Server Host Name (18 x 32 Bit)			
Bootfile Name (32 x 32 Bit)			
Vendor Specifics (16 x 32 Bit)			

Abb. 3–7 *Aufbau der BootP Message*

Die einzelnen Einträge haben folgende Bedeutung:

- *Operation* (8)

Wert 1 Boot-Request; Wert 2 Boot-Reply

- *Htype* (8)

Hardware-Adresstyp (z.B. Wert 1 für Ethernet)

- *Hlen* (8)
Länge der Hardwareadresse
- *Hops* (8)
Bei Clients wird hier der Wert 0 eingetragen. Wenn netzwerkübergreifende Boot-Vorgänge durchgeführt werden sollen, wird hier die Anzahl relevanter Hops (Netzwerkübergänge) eingetragen.
- *Transaction ID* (32)
Die nach dem Zufallsprinzip generierte Transaction-ID ist für die Zuordnung von Request und Reply zuständig.
- *Seconds* (16)
Hier wird vom Client die seit dem ersten Boot-Versuch verstrichene Zeit in Sekunden eingetragen.
- *Client IP-Address* (32)
IP-Adresse (Version 4) des Clients, wenn er diese kennt
- *Your IP-Address* (32)
IP-Adresse des Clients, wenn er diese nicht kennt (wird vom Server eingetragen)
- *Server IP-Address* (32)
IP-Adresse des Servers, die er hier in einen Boot-Reply einträgt
- *Gateway IP-Address* (32)
IP-Adresse des Gateways (bei Netzübergängen)
- *Client Hardware Address* (128)

Die vom Client eingetragene Hardwareadresse

- *Server Host Name* (512)

Dieser (optionale) Eintrag enthält den Host-Namen des Servers, der mit einer binären Null terminiert wird.

- *Bootfile Name* (1024)

Hier wird der NULL-terminierte Bootfile-Name eingetragen. Der Boot-Request enthält zunächst einen generischen Namen, wie UNIX oder DOS, um den betriebssystemspezifischen Reply generieren zu können. Der Reply enthält den vollqualifizierten Namen des Bootfiles (vollständige Pfadangabe).

- *Vendor Specifics* (512)

Ermöglicht die Angabe herstellerspezifischer Angaben.

Dynamic Host Configuration Protocol (DHCP)

Neben BootP (siehe vorhergehenden Abschnitt) stellt DHCP (*Dynamic Host Configuration Protocol*) einen weiteren Ansatz für eine dynamische Adressvergabe in einem IP-basierten Netzwerk dar. Dabei lässt sich festhalten, dass DHCP heutzutage das aktuellere Verfahren darstellt und auch an neue Technologien wie IPv6 angepasst wird.

HINWEIS

DHCP stellt eine für die Verwaltung eines IP-basierenden Netzwerks interessante Technologie dar, die in einer ersten Version im RFC 1541 vom Oktober 1993 beschrieben ist – die letzte ausführliche Beschreibung dazu findet sich im RFC8415 vom November 2018 (DHCP for IPv6) und wird DHCPv6 genannt.

Einführung

Der in den 90er Jahren des vorigen Jahrhunderts einsetzende »Internet-Boom« hatte den Nachteil, dass das zur Verfügung stehende Kontingent an öffentlich registrierbaren IP-Adressen bereits nach wenigen Jahren deutlich reduziert war. Das in Klassen eingeteilte IP-Adress- und Netzwerkkonzept bestand (unter IPv4) aus einem Adressraum von 32 Bits. Je nach Klassen- bzw. Subnetzeinteilung konnten innerhalb eines Subnetzes zwischen 254 bis 16,7 Millionen einzelne IP-Rechner adressiert werden. Dies schien auf den ersten Blick zwar eine unüberschaubar große Anzahl an Adressierungsmöglichkeiten, allerdings waren diejenigen Subnetze bereits seit Jahren vergeben, die eine hohe Anzahl von IP-Adressen bereithalten (Klasse-A-Adressen). Selbst Klasse-B-Adressen waren sehr schnell rar und eher selten zu bekommen. Als dann das Restkontingent von Klasse-C-Adressen ebenfalls zu versiegen

drohte, wurden neue Möglichkeiten entwickelt, um den unmittelbar bevorstehenden »Internet-GAU« abzuwenden.

Dazu wurden durch eine klassenübergreifende Neustrukturierung der IP-Adressen (*Classless Inter-Domain Routing* – CIDR) individuelle Adresskonzepte möglich, die den Anforderungen der Netzwerke gerecht werden konnten. Die nächste Entwicklungsstufe führte in eine flexiblere Adressgestaltung, die eine bedarfsorientierte Adressierung ermöglicht, bei der nicht jeder IP-Rechner von vornherein eine fest zugeordnete IP-Adresse erhält. Vielmehr wird ihm erst nach Einschalten und Zugriff auf das Netzwerk eine Adresse reserviert. Dabei geht dieses Verfahren von der Annahme aus, dass nicht alle Rechner eines Unternehmens gleichzeitig kommunizieren bzw. in Betrieb sind. Es reicht also aus, ein entsprechend verringertes Kontingent an Adressen vorzuhalten, um die erforderliche Kommunikationsleistung zu erreichen. Diese dynamische Adressvergabe wird durch das *Dynamic Host Configuration Protocol* (DHCP) realisiert (siehe [Abb. 3–8](#)).

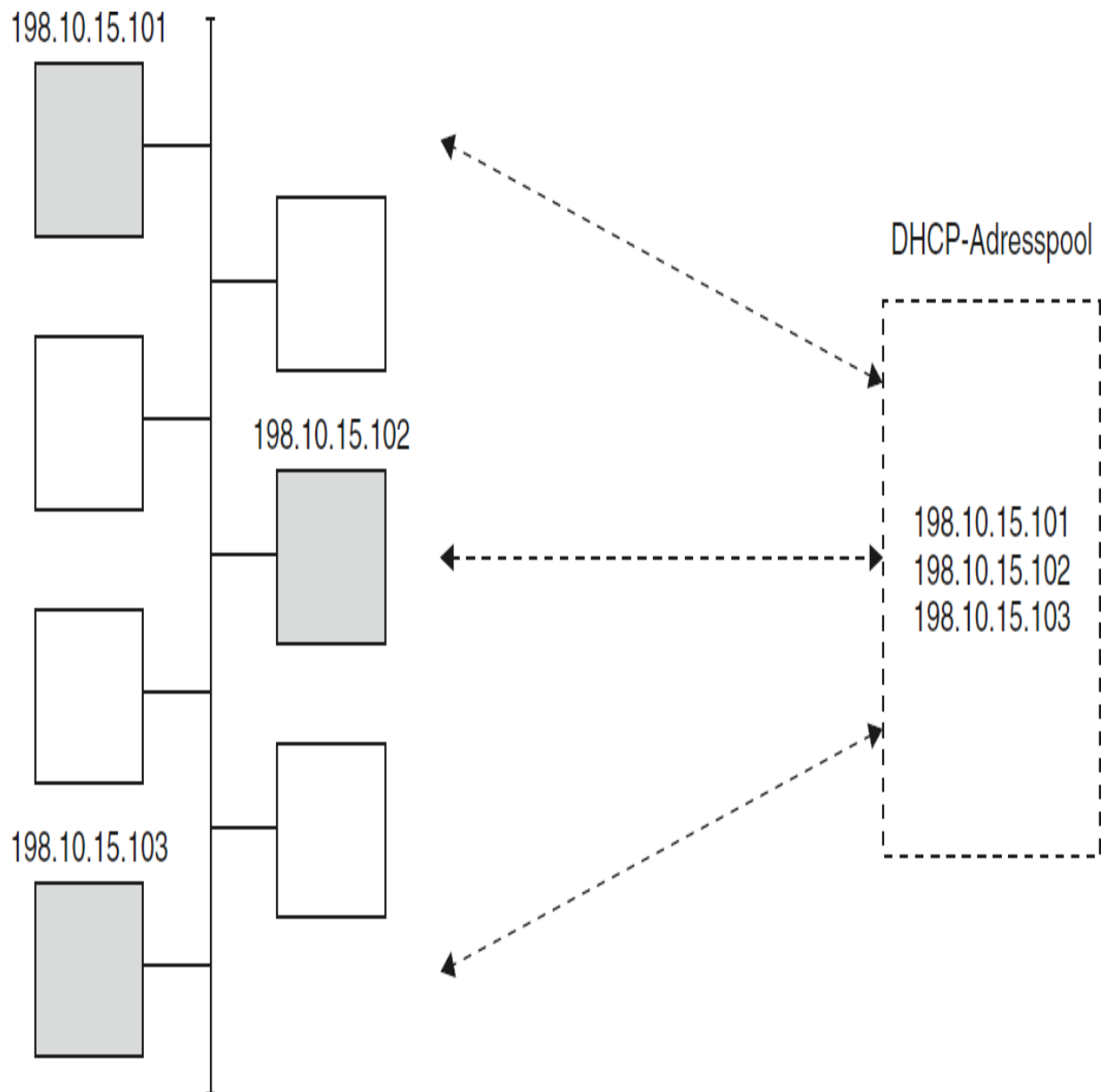


Abb. 3–8 *Prinzip des DHCP*

Einsatzzweck

Der Einsatz von DHCP kann für das interne Netzwerk eines Unternehmens oder einer Organisation vor allem administrative Vorteile haben. Für die Realisierung einer

umfassenden IP-Kommunikation müssen in mittleren bis großen Netzwerken oft mehrere Hundert bis Tausend IP-Adressen verwaltet werden. Dies bedeutet eine Vielzahl von Aktivitäten, die von mindestens einem Mitarbeiter in täglich anfallender Routine ausgeführt werden müssen. Dies sind im Einzelnen:

- Vergabe neuer IP-Adressen (Netzwerkerweiterung)
- Löschen von IP-Adressen (Ressourcenabbau)
- Anpassung bei Änderungen der Infrastruktur
- Netzwerkkonfiguration der IP-Clients

Sofern sich der Vorgang der Netzwerkanbindung weitgehend automatisieren lässt, kann ein Großteil der beschriebenen Aktivitäten entfallen. Dabei unterstützt DHCP drei Mechanismen zur IP-Adresszuordnung:

- *Automatischer Mechanismus*
Sorgt für eine permanente IP-Adresszuordnung.
- *Dynamischer Mechanismus*
Stellt einem IP-Host lediglich für einen begrenzten Zeitraum eine IP-Adresse zur Verfügung.
- *Manueller Mechanismus*
Erfordert die manuelle Zuordnung durch den Systemverwalter, wobei DHCP dabei lediglich für den

Transport der IP-Adresse benötigt wird.

Die Wahl des jeweiligen Verfahrens (oder von Kombinationen) wird nicht zuletzt durch Vorgaben oder Festlegungen in der Netzwerk- und Systemverwaltung bestimmt. Allerdings besitzt der *dynamische Mechanismus* eine besondere Bedeutung, da nur in dieser Variante durch die Wiederverwendung von temporär benutzten IP-Adressen eine optimierte Ressourcenverwaltung möglich ist.

Merkmale und Format

Das Format einer DHCP-Nachricht basiert auf Nachrichten des Bootstrap-Protokolls (BootP; siehe vorhergehender Abschnitt). Der kommunikationsbereite Rechner wird eingeschaltet und versendet im Netzwerk *BootP Requests*. Diese *Requests* werden vom zuständigen DHCP-Server registriert und mit der Zusendung von IP-Konfigurationsdaten beantwortet. Um nicht in jedem IP-Subnetz einen eigenen DHCP-Server installieren zu müssen, werden sogenannte *BootP Relay Agents* eingerichtet, die eine Weiterleitung der *BootP Requests* vornehmen. Auf der Ebene der Router werden zu diesem Zweck sogenannte *Helper-Adressen* konfiguriert, die eine Weiterleitung der sonst nicht übertragenen DHCP-Pakete ermöglichen.

DHCP wird dazu verwendet, IP-Hosts (keine Router) mit den zur Kommunikation erforderlichen Konfigurationsparametern zu versorgen (eigene IP-Adresse, Standard-Gateway usw.). Nachdem diese Parameter per DHCP empfangen wurden, ist ein IP-Host (z.B. Rechner) in der Lage, mit anderen Hosts Datenpakete auszutauschen. Es werden nicht immer sämtliche Parameter zur Host-Initialisierung benötigt; die tatsächlich erforderlichen Parameter werden vielmehr zwischen DHCP-Client und -Server ausgehandelt. Somit werden beim Einsatz von DHCP folgende Ziele verfolgt:

- Kontrolle der Konfigurationsparameter und lokaler Ressourcen
- keine manuelle Host-Konfiguration
- Vermeidung von DHCP-Serverinstallationen auf jedem einzelnen Netzwerk durch Einrichtung von BootP/DHCP Relay Agents
- DHCP-Clients müssen in der Lage sein, Antworten mehrerer DHCP-Server verarbeiten zu können.
- Koexistenz mit statisch konfigurierten IP-Hosts
- DHCP muss mit BootP Relay Agents zusammenarbeiten können.
- Bereits existierende BootP-Clients müssen durch DHCP ebenfalls bedient werden können.
- keine doppelte Vergabe von IP-Adressen

Protokollspezifikationen

Aus der Sicht eines Clients (IP-Hosts) stellt DHCP eine Erweiterung des Boot-Protokolls dar. Daher können auch bestehende BootP-Clients mit DHCP-Servern kommunizieren, ohne explizit dafür konfiguriert worden zu sein. [Abbildung 3–9](#) zeigt die Struktur einer DHCP-Nachricht. Der Aufbau von Frames bzw. Datenpaketen wird normalerweise vertikal zu jeweils 32 Bits skizziert (jedes Byte wird dabei auch *Oktett* genannt).

op	htype	hlen	hops
xid			
secs		flags	
ciaddr			
yiaddr			
siaddr			
giaddr			
chaddr (16)			
sname (64)			
bfile (128)			
options (312)			

Abb. 3–9 *Aufbau und Struktur einer DHCP-Nachricht*

Die einzelnen Felder der DHCP-Nachricht haben folgende Bedeutung (die Klammerwerte stellen die Feldgröße in Bit dar):

- *op (8)*
Angabe zur Art der Meldung (*message operation code/message type*; 1 = bootrequest, 2 = bootreply)

- *htype (8)*
Typ der Hardwareadresse (siehe RFC 1340 *Assigned Numbers*)
- *hlen (8)*
Länge der Hardwareadresse
- *hops (8)*
Bei Clients wird hier der Wert 0 eingetragen. Wenn über *Bootp Relay Agents* gebootet wird, so wird hier die Anzahl der Netzübergänge (Hops) eingetragen.
- *xid (32)*
Eine nach Zufallsprinzip generierte Transaktionsidentifikation sorgt für eindeutige Zuordnungen von *Request* und *Reply* innerhalb der Client-Server-Kommunikation.
- *secs (16)*
Hier wird vom Client die Zeit (in Sekunden) eingetragen, die seit seinem ersten Boot-Versuch verstrichen ist.
- *flags (16)*
Dieses Feld erhält vom Client im ersten Bit den Wert 1. Dieses Bit (*Broadcast Flag*) ist für die Auswertung durch DHCP-Server und *BootP Relay Agents* von Bedeutung. Alle weiteren Bits dieses Feldes erhalten den Wert 0.
- *ciaddr (32)*

IP-Adresse des Clients, die in einem DHCPREQUEST eingetragen wird, um zuvor übernommene Konfigurationsparameter zu verifizieren

- *yiaddr (32)*

Your (Client) IP-Address. Wenn der Client seine IP-Adresse nicht kennt, wird diese vom DHCP-Server hier eingetragen.

- *siaddr (32)*

IP-Adresse des nächsten DHCP-Servers, der in einen *bootreply* DHCPOFFER, DHCPACK oder DHCPNAK einträgt

- *giaddr (32)*

IP-Adresse des *BootP Relay Agent*, sofern über ihn gebootet werden soll

- *chaddr (128)*

Hardwareadresse des Clients

- *sname (512)*

Dieser (optionale) Eintrag enthält den DHCP-Servernamen und wird mit einem NULL-String terminiert.

- *file (1024)*

Hier wird der NULL-terminierte Bootfile-Name eingetragen. Der *bootrequest* (DHCPDISCOVER) enthält zunächst einen generischen Namen. In den *Reply* wird der (betriebssystemabhängige) vollqualifizierte Pfadname des Bootfiles eingetragen.

- *options (512)*

Angabe von Optionen (z.B. Herstellerangaben)

Der folgenden Aufstellung sind jeweils die einzelnen DHCP-Nachrichtentypen zu entnehmen:

Nachrichtentyp	Beschreibung
DHCPDISCOVER	Client-Broadcast zur Lokalisierung verfügbarer Server
DHCPOFFER	Antwort des DHCP-Servers auf ein DHCPDISCOVER mit der Angabe von Konfigurationsparametern
DHCPREQUEST	Client-Broadcast an DHCP-Server zur Anforderung angebotener Parameter von einem dedizierten Server bei gleichzeitiger Ablehnung der angebotenen Parameter aller anderen Server
DHCPACK	Server schickt Client-Konfigurationsparameter einschließlich der festgelegten IP-Adresse.
DHCPNAK	Server lehnt Konfigurationsparameter-Anforderung (auch IP-Adresse) des Clients ab.
DHCPDECLINE	Client informiert Server, dass Konfigurationsparameter bzw. IP-Adresse ungültig sind.

DHCPRELEASE Client informiert Server, dass er nun die IP-Adresse nicht mehr benötigt.

Funktionsweise

Anhand eines Beispiels soll das DHCP-Verfahren erläutert werden. Es wird beschrieben, wie eine Netzwerkadresse in sechs Phasen dem anfragenden DHCP-Client zugeordnet wird.

- *Phase 1*

Der Client versendet innerhalb seines Subnetzes (Subnet-Broadcast) eine DHCPDISCOVER-Nachricht. Diese enthält ggf. Optionen für einen IP-Adressvorschlag und einen Zuordnungszeitraum (im Originaltext wird dieser *Lease* genannt). BootP Relay Agents reichen diese Nachricht nicht nur an DHCP-Server innerhalb, sondern auch außerhalb des lokalen Subnetzes weiter.

- *Phase 2*

Jeder verfügbare DHCP-Server antwortet mit einer DHCPOFFER-Nachricht, die entweder direkt oder per *Subnet-Broadcast* an den Client übertragen wird. Die Antwort enthält eine verfügbare Netzwerkadresse und weitere Konfigurationsparameter in den DHCP-Optionen.

- *Phase 3*

Der Client erhält eine oder mehrere DHCPOFFER-Nachrichten von einem oder mehreren Servern. Er darf allerdings auf mehrere Antworten warten, bevor er sich an einen konkreten Server wendet, von dem er dann die Konfigurationsparameter gemäß DHCPOFFER-Nachricht anfordert. Nun verschickt der Client eine DHCPREQUEST-Nachricht per *Broadcast*, die den *server identifier* als Option beinhalten muss. Dies ist erforderlich, damit der vom Client ausgewählte Server exakt identifiziert werden kann. Weitere Optionen mit Angaben zu Konfigurationswerten können ebenfalls eingetragen werden. Diese DHCPREQUEST-Nachricht wird an BootP Relay Agents weitergeleitet. Um sicherzustellen, dass die Relay Agents die Nachricht an dieselben Server weiterleiten, die auch die DHCPDISCOVER-Nachricht erhalten haben, muss die DHCPREQUEST-Nachricht denselben Wert im *secs*-Feld erhalten und dieselbe Broadcast-Adresse verwenden. Wenn der Client keine DHCPOFFER-Nachricht erhält, läuft ein Timer ab und sendet die DHCPDISCOVER-Nachricht erneut.

- *Phase 4*

Die Server empfangen vom Client den DHCPREQUEST-Broadcast. Diejenigen Server, die vom Client nicht selektiert worden sind (Eintrag *server identifier*), betrachten die DHCPREQUEST-Nachricht als Ablehnung. Der selektierte

Server verpflichtet sich nun, dem Client Ressourcen bereitzustellen, und antwortet mit einer DHCPACK-Nachricht mit Angabe der Konfigurationsparameter des anfragenden Clients. Die Kombination aus *chaddr* (*Client Hardware Address*) und einer IP-Adresse bildet eine eindeutige Identifikation innerhalb des Zuordnungszeitraums für jeden DHCP-Nachrichtenverkehr. In das Feld *yiaddr* (*your IP-Address*) der DHCPACK-Nachricht wird die zugeordnete IP-Adresse eingetragen.

HINWEIS

Sollte der selektierte Server jedoch nicht in der Lage sein, den DHCPREQUEST zu beantworten (wenn beispielsweise die angeforderte Netzwerkadresse bereits vergeben wurde), wird der Server mit einer DHCPNAK-Nachricht antworten.

- *Phase 5*

In Phase 5 erhält der Client die DHCPACK-Nachricht samt Konfigurationsparametern, führt einen abschließenden Test der Parameter durch (z.B. ARP für die zugeordnete IP-Adresse) und notiert die Zuordnungsdauer und die spezifizierte Identifikation. Die Konfiguration des Clients ist damit abgeschlossen. Stellt der Client jedoch fehlerhafte

Parameter in der DHCPACK-Nachricht fest, übermittelt er dem Server eine DHCPDECLINE-Nachricht und wiederholt den Konfigurationsprozess. Der Client sollte mindestens 10 Sekunden warten, bevor er diesen Prozess startet, um unnötigen Netzwerkverkehr zu vermeiden. Der Client wiederholt den Konfigurationsprozess ebenfalls, wenn er eine DHCPNAK-Nachricht erhält. Sollte er in einem festgelegten Zeitintervall weder DHCPACK- noch DHCPNAK-Nachrichten empfangen, so wiederholt er die DHCPREQUEST-Nachricht. Wenn der Client nach zehn Wiederholungen der DHCPREQUEST-Nachricht weder eine DHCPACK- noch eine DHCPNAK-Nachricht erhalten hat, erfolgt eine vollständige Initialisierung des Prozesses. Außerdem wird der Anwender über diesen Status informiert.

- *Phase 6*

Wenn der Client seine Adresszuordnung beenden will, sendet er dem Server eine DHCPRELEASE-Nachricht. Der Client identifiziert die aufzugebende Zuordnung durch den Eintrag seiner IP-Adresse ins Feld *ciaddr* (*Client IP-Address*) und seiner Hardwareadresse ins Feld *chaddr* (*Client Hardware Address*).

IP-Version 6 (IPv6)

Bereits seit Mitte der 90er Jahre des vorigen Jahrhunderts ist es kein Geheimnis mehr, dass IP-Adressen ein knappes Gut sind. So wurden damals in erster Näherung die Anfang der 80er Jahre festgelegten Richtlinien zum *Internet Protocol* überarbeitet, um den Anforderungen eines modernen und zukunftsweisenden (routbaren) Übertragungsprotokolls gerecht werden zu können. In der Zwischenzeit konnte sich das Internetprotokoll mehr als vierzig Jahre lang in der zurzeit (immer noch) aktuellen Version 4 behaupten. Die Situation hätte sich wohl auch so schnell nicht geändert, wenn nicht seit einigen Jahren der Bedarf an IP-Adressen so enorm gestiegen wäre, da immer mehr Geräte in Netzwerke eingebunden werden (siehe auch Abschnitt 9.1 »Internet of Things« – IoT). Die Verknappung ergibt sich einzig und allein aus der endlichen Verfügbarkeit der Adressen. IP-Adressen (Version 4) sind 32 Bits lang und ermöglichen somit maximal 2^{32} (= 4.294.967.296) verfügbare IP-Adressen.

Dramatisiert wurde und wird diese Entwicklung durch die Tatsache, dass sich die Anzahl der Internetnutzer ca. alle ein bis anderthalb Jahre verdoppelt, wodurch in der Regel auch immer neue (offizielle) IP-Adressen benötigt werden. Eine Sättigung des »IP-Marktes« ist auch noch nicht absehbar – ganz im

Gegenteil. Vor allem im privaten Anwendungsbereich sind die Wachstumszahlen an IP-Adressen bzw. Zugängen ins Internet enorm gestiegen. Der Handel mit Internetdomänen boomt wie nie zuvor. Neben zahlreichen Unternehmen sind es zunehmend Privathaushalte, die einen Großteil der verfügbaren IP-Adressen für ihre Internet-Homepages beanspruchen.

Es ist aber nicht nur die rapide abnehmende Verfügbarkeit (offizieller) IP-Adressen, die bereits seit Jahren die Forderung nach einer neuen IP-Version laut werden lassen. Auch die veränderten Übertragungsformen in Bezug auf Echtzeitanwendungen, Multimedia-Übertragungen usw. sowie die Forderung nach mehr Sicherheit (z.B. im Bereich E-Commerce) führten dazu, dass sich die dafür eingesetzten Gremien bereits seit Jahren mit der Einführung einer neuen IP-Version (*IPv6*) beschäftigen.

HINWEIS

Da der IP-Adressraum und damit die zur Verfügung stehenden (offiziellen) Adressen unter IP-Version 4 immer knapper werden, wurde bereits vor Jahren mit der Entwicklung des IPv6-Standards begonnen.

Gründe für eine Neuentwicklung

Das Internet ist ein völlig anderes Medium (Netzwerk) als noch zu Beginn der 80er oder 90er Jahre des vorigen Jahrhunderts. Es ist mittlerweile das größte öffentliche Datennetz der Welt, und seine Ausbreitung wächst permanent. Dies ist bedingt durch die große Popularität des World Wide Web (WWW) und durch die Möglichkeiten, die Geschäftsleute sehen, ihre Kunden an einer *virtuellen Theke* bedienen zu können (*E-Commerce*). So lässt sich die Entwicklung des Internets mit den teilweise althergebrachten Ansätzen zur Ausweitung eines solchen Mediums nicht erklären. Bei aller zu Recht angebrachten Skepsis ist das Internet (in erster Linie ist auch hier wieder das WWW gemeint) das Übertragungsmedium der Zukunft.

Da der offizielle IP-Adressbestand mehr und mehr aufgebraucht wurde, lag der Hauptgrund für den Bedarf einer Änderung an der IP-Version 4 in einem vorhersehbaren Mangel an verfügbaren IP-Adressen. Damit verbunden waren aber immer auch die Wünsche nach Optimierung des mittlerweile sehr betagten IP-Protokolls. So sind insbesondere auch die Bedenken über das Anwachsen der benötigten Verwaltungsinformationen in den Netzknoten (Router), die für die Weiterleitung und Verteilung der Daten zuständig sind, sehr leicht nachzuvollziehen.

Das Problem der unzureichenden Anzahl von Adressen wurde noch durch die sehr ineffiziente Vergabe von (offiziellen) IP-Adressen verstärkt. Der Adressraum schien bei der Entwicklung von IP-Version 4 so groß, dass eine Aufteilung in Adressklassen am sinnvollsten erschien. Die Routing-Tabellen werden dadurch klein gehalten, da pro Netz nur ein Eintrag erforderlich ist. In der Praxis schränkt dieser Mechanismus jedoch die zur Verfügung stehenden Adressräume unnötig ein, denn durch die klassenweise Nutzung der Adressen wird der zur Verfügung stehende Adressraum nicht vollständig genutzt.

Darüber hinaus wurden in der Vergangenheit Firmen Adressen aus dem Klasse-B-Bereich zugewiesen, obwohl diese vielleicht maximal 2000 Adressen benötigten. Ein Klasse-B-Netz verfügt jedoch über maximal 65.534 Adressen, womit in diesem Fall über 63.000 Adressen verloren wären. In der Praxis zeigt sich immer wieder, dass beispielsweise an den Klasse-B-Netzen wesentlich weniger Endgeräte angeschlossen sind, als eigentlich möglich wären. Bei Untersuchungen in Deutschland wurde festgestellt, dass durchschnittlich weniger als 2.500 Rechner an diesen Netzen angeschlossen sind. In der Konsequenz bedeutet dies, dass ein Großteil des zur Verfügung stehenden Adressraums verschenkt wird.

Da die Wachstumsrate des Internets jedoch permanent weiter ansteigt und Klasse-B-Adressen kaum noch verfügbar sind, entschlossen sich die für die Adressvergabe verantwortlichen Behörden, nur noch Blöcke von Klasse-C-Adressen zu vergeben. Der Nachteil dieser Verfahrensweise ist jedoch, dass für jedes dieser Klasse-C-Netze ein eigener Eintrag in den Routing-Tabellen notwendig ist, wodurch diese stark anwachsen und die Router überdurchschnittlich belasten. Das Problem der aufwendigen Verwaltung der Routing-Informationen wird durch dieses schnelle Wachstum verursacht. Die zentralen Router sollten nach Möglichkeit vollständige Routing-Angaben über das Internet besitzen. Bedingt durch den Anschluss vieler Unternehmen und Organisationen an das Internet hat dies jedoch zu einem explosionsartigen Anwachsen der Routing-Tabellen geführt. Dabei ist zu bedenken, dass das Routing-Problem nicht einfach dadurch gelöst werden kann, dass in den Routern mehr Arbeitsspeicher installiert wird und die Routing-Tabellen vergrößert werden. Andere Faktoren der Probleme in den zentralen Verteilern sind die notwendige CPU-Leistung, um Änderungen der Tabellen und der Topologie nachvollziehen zu können. Die dynamische Natur des WWW und ihre Einflüsse auf die Cache-Speicher der Router tun ihr Übriges dazu. Sofern die Routing-Tabellen in den zentralen Routern beliebig

wachsen sollen, sind die Router irgendwann gezwungen, Routen zu vergessen, womit Teile des Internets nicht mehr erreichbar wären.

HINWEIS

Nähere Angaben zum Thema »Routing« und dessen Besonderheiten enthält [Kapitel 4](#).

Neben den allgemeinen Erfordernissen einer neuen IP-Version stellt natürlich auch die Entwicklung der zu übertragenden Daten mittlerweile neue Anforderungen an das Übertragungsprotokoll. Speziell die Entwicklung von Techniken bzw. Technologien wie beispielsweise Multimedia, Echtzeitanwendungen, Gebäudeleitsystemen, mobilen Systemen, TV-Empfangsgeräten neuer Generation, Haushaltsgeräten mit Internetanschluss oder auch die IP-Telefonie (*Voice over IP*) erfordern eine viel flexiblere Möglichkeit der Datenübertragung.

Die wesentlichen Anforderungen an eine neue Version des IP-Übertragungsprotokolls lassen sich generell wie folgt zusammenfassen:

- deutliche Vergrößerung des verfügbaren Adressraums
- Einführung von Sicherheits-Dienstelementen

- Unterstützung von Multicasting
- Bildung und Auswertung von Dienstgüteklassen zur Unterstützung von Multimedia-Anwendungen
- Synchronisation von Datenströmen
- Reservierung von Ressourcen
- Unterstützung mobiler Systeme
- Verhinderung von Überlastsituationen
- Verrechnung bezogener Leistungen

Kurzfristige Maßnahmen wie das *Classless Inter-Domain Routing* (CIDR), das eine bessere Aufteilung der IP-Klassengesellschaft ermöglichte, oder aber neue Verfahren beim Web-Hosting (mehrere Identitäten für eine einzige IP-Adresse) konnten bestenfalls das Leben der IP-Version 4 um einige Jahre verlängern. Es ist allerdings nicht nur der private Internetkunde, der für eine Adressknappheit gesorgt hat. Sehr viele Unternehmen, die in den letzten Jahren ein heterogenes Rechner- und Netzwerkkumfeld entstehen ließen, bedienen sich mittlerweile des Netzwerkprotokolls TCP/IP und seiner integrativen Eigenschaften. In dieser Situation ergibt sich grundsätzlich die Möglichkeit, entweder ein IP-Netzwerk ohne registrierte IP-Adressen zu etablieren, d.h., eine Netzanbindung an das öffentliche Internet darf nicht vorgenommen werden, oder aber eine öffentliche IP-Adresse bzw. einen IP-Adressbereich für den geplanten Netzzugang zu beantragen. In

diesem Fall werden natürlich Adressressourcen verbraucht, die letztlich zur Weiterentwicklung der IP-Version 4 zur Version 6 beigetragen haben.

HINWEIS

Oft taucht die Frage auf, ob es keine Version 5 des IP-Protokolls gibt. Tatsächlich hat es ein Protokoll mit dem Namen IPv5 nie gegeben. Allerdings hatte die IANA die IP-Versionsnummer 5 für das *Internet Stream Protocol Version 2* reserviert (ST2, definiert in RFC 1819), das gegenüber IPv4 verbesserte Echtzeitfähigkeiten besitzen sollte, dessen Weiterentwicklung dann aber aus Kosten-Nutzen-Erwägungen zugunsten von IPv6 eingestellt wurde.

Lösungsansätze

Bei Betrachtung möglicher Lösungsvorschläge für eine neue IP-Version wurde immer wieder deutlich, dass die wichtigste Anforderung an eine neue IP-Version darin bestand, die Anzahl möglicher Adressen durch Erweiterung der Adressbits massiv zu erhöhen. Denn durch die schnelle Verbreitung des TCP/IP-Protokolls und das explosionsartige Wachstum des Internets sind Probleme entstanden, die bei der Entwicklung des ursprünglichen TCP/IP-Protokollstacks nicht absehbar waren.

Den Urhebern der IP-Spezifikationen schien eine 32-Bit-Adresse als absolut ausreichend. Zum damaligen Zeitpunkt konnte sich niemand vorstellen, dass jemals auf fast jedem Schreibtisch ein IP-Arbeitsplatz stehen würde.

Ein möglicher Ausweg aus der Adressverknappung besteht darin, große zusammenhängende Blöcke von Klasse-C-Adressen an Internet-Dienstleister (ISPs) abzutreten. Diese bieten dann ihrerseits den Kunden Blöcke von Klasse-C-Adressen an. Hierdurch wird das Problem der Adressverknappung kurzfristig umgangen. Wie bereits oben dargestellt, ist dies jedoch mit dem Nachteil verbunden, dass die Routing-Tabellen sehr schnell anwachsen.

Um das Problem der anwachsenden Routing-Tabellen zu lösen, wird dann für jeden Adressblock nur noch eine Zieladresse in der Routing-Tabelle eingetragen. Hauptproblem hierbei ist, dass diese Adresse als klassenlos interpretiert werden muss. Ein Router muss dann seine Wegewahl anhand des Vergleichs einer solchen (klassenlosen) Adresse mit der zugehörigen Subnetzmaske treffen. Der Vorteil des als *Supernetting* bezeichneten Verfahrens besteht darin, dass damit das Wachstum der Routing-Tabellen begrenzt und gleichzeitig auch größeren Netzen ein ausreichender Adressraum zur Verfügung gestellt werden kann. *Supernetting* wird im Internet

eingesetzt, um die Problematik der Adressraumverknappung zumindest für einen kurzen Zeitraum zu entschärfen.

HINWEIS

Supernetting kann die grundsätzliche Problematik der Adressverknappung, die die Beschränkung auf 32-Bit-Adressen mit sich bringt, nur übergangsweise lösen.

Die langfristige Lösung dieser Probleme ist grundsätzlich nur in der allgemeinen Verwendung von IPv6 zu sehen. Während das Internet auf IPv6 wartet, muss IPv4 so geändert werden, dass das Internet die benötigten Anbindungen zur Verfügung stellen kann. Dies führt unweigerlich zu Veränderungsprozessen, die nicht nur Schwierigkeiten verursachen, sondern auch fundamentale Konzepte des Internets ändern können.

Lösungen auf Basis von IPv6

Nicht zuletzt ausgelöst durch die verschiedenen Bestrebungen und Entwicklungen der diversen Arbeitsgruppen ist die neue IP-Version unter der Bezeichnung *IP-Version 6 (IPv6)* verfügbar.

Erste Überlegungen zum »IP der nächsten Generation« wurden bereits im Jahre 1992 angestellt, als vier unterschiedliche Vorschläge zum Thema IPNG (*IP Next*

Generation) vorgestellt wurden. Begriffe wie *CNAT*, *IP Encaps*, *Nimrod* oder *Simple CLNP* »geisterten« durch die Reihen der IETF-Arbeitsgruppen. Es folgten *PIP* (*P Internet Protocol*), *SIP* (*The Simple Internet Protocol*) und *TP/IX*. Eine weitere Entwicklung vollzog sich wenig später vom *Simple CLNP* zum *TCP and UDP with Bigger Addresses (TUBA)*. Die *IP Address Encapsulation (IPAE)* ging aus *IP Encaps* hervor. Aus diesen recht intensiven Bemühungen um eine rasche Klärung der mittlerweile kaum noch zu bewältigenden Probleme mit der alten IP-Philosophie gingen mittlerweile etwa siebzig RFCs hervor, die ein breites Spektrum an Problemstellungen behandeln. Im RFC 2460 vom Dezember 1998 wurde das neue Protokoll in den Status des »Draft Standard« gesetzt; mit RFC 8200 vom Juli 2017 wurde es zum Standard erklärt.

HINWEIS

Mittlerweile existieren bereits zahlreiche Systemimplementierungen, die IPv6 unterstützen. Dies gilt für fast alle Systemplattformen sowie auch nahezu sämtliche am Markt erhältlichen Netzwerkkomponenten (z.B. Router, Switches).

Die gesamte Entwicklung der IP-Version 6 ist auf Technologietrends im Bereich der Vernetzung ausgerichtet. So

ist der vereinfachte Protokollheader für eine schnelle, eventuell hardwareinterne Verarbeitung in Hochgeschwindigkeitsprotokollen vorbereitet. Auch das Fehlen einer Checksumme ist für eine schnelle Verarbeitung hilfreich. Die Einführung des Flow Label macht es schließlich möglich, zusammenhängende Datenflüsse zu realisieren, die spezielle Anforderungen beispielsweise an Latenzzeiten oder den Datendurchsatz stellen. Solche Mechanismen werden z.B. bei der Übertragung von Sprachdaten oder Videos benötigt.

Bei IPv6 sind auch die Broadcast-Adressen weggefallen. An ihre Stelle tritt eine Multicast-Adressierung (Gruppen-Adressierung), die die Funktionen des Broadcast als Spezialfall beinhaltet. Bei den an einen einzelnen Empfänger gerichteten Unicast-Adressen sind folgende Varianten vorgesehen:

- hierarchische Dienstanbieter-Adressen
- hierarchische geografische Adressen
- NSAP-, IPX- und Cluster-Adressen
- IP-Only- und IPv4-kompatible SIPP-Adressen

ROAD-Arbeitsgruppe

Ausgehend von den Überlegungen zur Schaffung einer neuen IP-Version wurde im Jahre 1992 eine spezielle Arbeitsgruppe mit dem Namen *ROAD* gegründet. Dabei steht ROAD für

ROuting and ADdressing. Diese Arbeitsgruppe wurde von der *Internet Engineering Task Force (IETF)* gegründet und die wesentlichen Aufgaben sind wie folgt definiert:

- Erarbeitung und Einführung einer größeren und klassenlosen Struktur für IP-Adressen
- Festlegung von Verfahren, die die Router entlasten und das Wachstum der Routing-Tabellen begrenzen
- Neue Adressstrukturen sollen möglichst nur auf den zentralen Routern implementiert werden, sodass existierende Endgeräte nicht modifiziert werden müssen.

Die ROAD-Gruppe hat unterschiedliche Lösungsansätze erarbeitet, denen verschiedene Bezeichnungen zugewiesen wurden und die nachfolgend erläutert werden.

SIPP

Bei dieser angedachten Lösung (*SIPP = Simple Internet Protocol Plus*) wird das 32-Bit-IP-Adressfeld beibehalten und die Adresslänge nicht verändert. Die IP-Adresse reduziert sich von einer weltweit gültigen auf eine nur lokal gültige Adresse (innerhalb einer sogenannten *Administration Domain*). Die Umsetzung auf ein globales Adressschema erfolgt in den Routern (Gateways), die die Verwaltungsdomäne mit dem Rest der Welt verbinden.

Dieser Ansatz hat den Vorteil, dass die heute gültige IP-Software der Rechner in den Verwaltungsdomänen unverändert beibehalten werden kann. Nur die Datenpakete, die die Verwaltungsdomäne verlassen, müssen auf ein neues Adressierungsverfahren umgesetzt werden; dieses Verfahren ist auch als *Network Address Translation (NAT)* bekannt.

PIP

Bei dieser Form werden die IP-Adressen auf ein Format mit 64 Bits (oder größer) vergrößert. Das neue Format enthält dabei neben der global einzigartigen Rechneradresse noch einen sogenannten *Administration Domain Identifier (ADI)*. Dies hat zur Folge, dass die heute eingesetzte IP-Software vollkommen geändert werden muss. Das Routing zwischen den einzelnen Domains (Domänen) erfolgt nach den gleichen Mechanismen wie bisher.

IP over IP

Beim Ansatz *IP over IP* werden die 32-Bit-IP-Adressen beibehalten und die Adresslänge wird nicht verändert. Bei einem Verbindungsaufbau zu Rechnern außerhalb der eigenen Administration Domain (über Router) wird das Original-IP-Paket um einen weiteren *IP-Header* erweitert. Dieser zusätzliche Header enthält das um den *Administration Domain*

Identifiziert erweiterte Adressschema. Bekannt geworden ist dieses Verfahren auch unter der Bezeichnung ENCAPS (*Encapsulation* = Einkapselung).

TUBA

Unter dem Arbeitstitel *TCP and UDP with Bigger Addresses* (TUBA) wurde ein weiterer Lösungsvorschlag entwickelt. Bei TUBA werden alle TCP- und UDP-Anwendungen mittel- bis langfristig auf Basis des *Connectionless Network Layer Protocol* (CLNP) portiert. TUBA definiert sowohl die Funktionen der Hilfsprotokolle auf den beiden Schichten *Network-/Internet-Layer* als auch die Struktur der unter OSI eingesetzten hierarchischen NSAP-Adressen (*Network Service Access Point*).

HINWEIS

TUBA ermöglicht grundsätzlich eine sanfte Migration, da über einen kurz- bis mittelfristigen Zeitraum das bisher verwendete IP-Adressschema parallel verwendet werden kann.

Die Experten der ROAD-Arbeitsgruppe hatten als Zwischenlösung bis zur Verabschiedung eines IP-Standards eine neue Einteilung der bisherigen Adressen vorgeschlagen. Dabei blieben die Klasse-A- und Klasse-B-Adressen unberührt; nur die Klasse-C-Adressen wurden in kleinere Einheiten zerlegt, sodass

noch mehr IP-Netze mit weniger Hosts definiert werden konnten. Als weiterer Ansatz diente die Aufteilung des Klasse-E-Bereichs in weitere Adressklassen (Klasse F, G, H und K).

IPv6-Leistungsmerkmale

Antriebsmotor für die Entwicklung der neuen IP-Version 6 war sicherlich die trotz einiger Gegenmaßnahmen (z.B. CIDR) nicht vermeidbare Adressknappheit. Sich neu entwickelnde Märkte, vor allem im Consumer-Segment, lassen einen stetig wachsenden Bedarf an adressierbaren Netzwerkkomponenten erwarten. Es ist dabei nicht nur daran zu denken, dass sich Anforderungen zur Vernetzung von mittlerweile in fast jedem Haushalt installierten Personalcomputern oder an verschiedenen Plätzen einsetzbaren mobilen Computern und sonstigen Geräten (Smartphone, PDA usw.) ergeben, sondern es zeigt sich bereits teilweise die Notwendigkeit, Fernsehgeräte, Receiver oder Aufzeichnungsgeräte in ein globales Netz zu integrieren. Diese Vernetzung der Haushalte kann allerdings nur über ein universell einsetzbares Netzwerkprotokoll vorgenommen werden, da es um die kommunikative Zusammenführung unterschiedlichster Hardware geht. Solchen Anforderungen muss IPv6 gerecht werden können. Aus all diesen und weiteren Überlegungen hat sich ein Spektrum von

Leistungsmerkmalen herausgebildet, das die wesentlichen Eigenschaften des IPv6 charakterisiert.

Erweiterung des Adressraums

Die bisher gültige Adressierung im IP-Netzwerk wurde durch eine 32 Bits umfassende IP-Adresse realisiert. Die vier bzw. fünf Klassen dieses Konzepts sind jedoch mittlerweile ausgeschöpft, und so wurde eine Erweiterung des Adressraums auf 128 Bits vorgenommen. Die Anzahl verfügbarer IP-Adressen würde vermutlich ausreichen, um alle Planeten unserer Galaxie (bzw. ihre Bewohner) mit entsprechenden Adressen auszustatten. Die Schreibweise ändert sich jedoch von der altbekannten interpunktierten Notation (Beispiel 191.79.14.12) in eine byteweise Aufteilung, die durch Doppelpunkte getrennt ist (Beispiel: 2001:0db8:85a3:08d3:1419: 8a2e:0370:7544). Zur Wahrung der Kompatibilität zwischen IP-Version 4 und Version 6 werden die alten Adressen in den unteren Bereich der V6-Adressen eingeblendet. Etwa 15 % des definierbaren Adressraums sind bereits für bestimmte Service-Provider (Provider-ID) oder auch Fremdprotokolle (z.B. für OSI-Protokolle und IPX) reserviert. Der verbleibende Rest von 85% steht für die Implementierung der Netzwerkadressen zur Verfügung.

Abbildung von Hierarchien

Der sehr große Adressbereich ermöglicht eine nahezu unbegrenzte Abbildung hierarchischer Strukturen. Für die Darstellung auf Byte-Ebene lassen sich bereits 16 verschachtelte Hierarchien bilden. Werden diese Byte-Strukturen aufgebrochen und weitere Subhierarchien eingeführt (ähnlich den heutigen Subnetzen), so kann die Hierarchiebildung nahezu beliebig ausgebaut werden.

IP-Header-Struktur

Der (ohne Optionen) 20 Bytes umfassende IP-Version-4-Header dehnt sich, trotz Vergrößerung der IP-Adresslänge um den Faktor 4, lediglich um das Doppelte aus. Der neue IP-Version-6-Header ist zudem einfacher geworden, da einige Felder gegenüber der alten Version weggelassen worden sind.

Priorisierung

Ähnlich dem QoS (*Quality of Service*) in ISO/OSI-Protokollen (Beispiel ist hier das IS-IS-Protokoll) lassen sich nun auch IP-Pakete mit besonderer Kennzeichnung versehen, die zu einer unterschiedlich gewichteten Behandlung während des Netztransports führt. Echtzeitdaten (z.B. Bildfolgen aus

Videokonferenzen) können somit wesentlich schneller als Daten für den normalen Online-Betrieb transportiert werden.

Sicherheit

Ein wichtiger Auslöser für die Neuausrichtung der Internetprotokolle waren auch die Sicherheitsmerkmale (*Chiffrierung* bzw. *Authentisierung*) und die Qualitätsanforderungen oder Dienstgüte (*Flow Labeling*), die durch die geschäftliche wie private Nutzung des Internets entstehen und im ursprünglich für die Forschung und Lehre entwickelten Netzwerk zunächst von geringem Interesse waren. IPv6 stellt somit vereinfachte und von den Netzknoten schneller verarbeitbare Protokollformate, eine optimierte Wegewahl (Routing) sowie eine verbesserte Unterstützung für spätere Erweiterungen und Optionen zur Verfügung.

Die bisher in der Anwendungsschicht vorgenommenen Authentifizierungsmechanismen werden mit IP-Version 6 auf der Netzwerkschicht übernommen. Es können zwei verschiedene Mechanismen verwendet werden. Der *IPv6 Authentication Header* stellt eine Option zur Authentifizierung von IP-Datagrammen und ihrer Integrität dar. Zwei Kommunikationspartner müssen sich gegenseitig ausweisen (Echtheitsprüfung), um miteinander kommunizieren zu

können. Der entsprechende Algorithmus wird mit MD5 bezeichnet. Eine weitere Option wird durch den *IPv6 Encapsulation Security Header* beschrieben. Hier wird die Vertraulichkeit von Daten mithilfe des DES-Algorithmus garantiert.

Ein äußerst wichtiger Aspekt bei der Entwicklung und Einführung des IPv6 ist die Fähigkeit zum Parallelbetrieb. IP-Netzwerke nach alter IP-Version 4 müssen problemlos in die Version 6 überführt werden können. Die Akzeptanz des neuen Standards hängt dabei immer ganz wesentlich von der Kompatibilität ab.

Vereinfachte Konfiguration

Bei der Entwicklung von IPv6 wurde ein Schwerpunkt auf eine stark vereinfachte und automatisierbare Konfiguration gesetzt. Insbesondere für einen Systemverwalter ist die schnellere Konfigurationsmöglichkeit eine der wesentlichen Neuerungen von IPv6. So muss beispielsweise bei IPv4 den einzelnen Endgeräten auf aufwendige Art und Weise eine eindeutige IP-Adresse zugewiesen werden. Nicht selten kommt dafür ein sogenannter DHCP-Server (*Dynamic Host Configuration Protocol*) zum Einsatz, der jedem System nach dem Start aus einem vordefinierten Wertebereich automatisch eine IP-

Adresse zuweist. Bei IPv6 erkennt ein startender Rechner das Netzwerk, in dem er sich befindet, automatisch und kann sich selbst konfigurieren. Ein manuelles Eingreifen des Systemverwalters kann entfallen. Aber auch unter IPv6 kann nach wie vor ein DHCP-Server eingesetzt oder eine Kombination von statischer und dynamischer Adressierung konfiguriert werden.

Multicasting

Eine weitere Forderung bei der Entwicklung einer neuen IP-Version war der Einsatz einer Multicasting-Funktion. Multicasting ist bei IPv6 als Basisfunktion implementiert und ermöglicht einem System, zur gleichen Zeit mit einer ganzen Gruppe anderer Systeme zu kommunizieren, die sich mittels einer gemeinsamen Multicast-Gruppenadresse als Gruppe eingetragen hat. Anwendungen, die davon Gebrauch machen können, sind zum Beispiel Audio- oder Videokonferenzen.

HINWEIS

Während sich *Multicasting* nur auf eine begrenzte (definierte) Anzahl von Systemen beschränkt, erreicht *Broadcasting* alle Systeme eines Netzwerks.

IP-Header der Version 6

Bei der Entwicklung von IPv6 wurde das bestehende Header-Format zwecks flexiblerer Erweiterungsmöglichkeiten vereinfacht. Zudem ist die Möglichkeit vorgesehen, die einzelnen Pakete eines Datenstroms als zusammenhängenden Fluss zu markieren (*Flow Label*). Der IP-Header von IPv6 stellt insgesamt 16 Bytes für die IP-Nummern zur Verfügung und ist durch einen vereinfachten Aufbau gekennzeichnet. Damit ist IPv6 auf zukünftige Erfordernisse vorbereitet, da Erweiterungen leicht realisierbar sind. Einige Header-Felder wurden bereits in IPv4 nur sehr selten genutzt. Um die für die Übermittlung der IP-Daten notwendige Bandbreite so gering wie möglich zu halten, wurden in der neuen IP-Version alle unnötigen Header-Felder eliminiert. Folgende Vorteile ergeben sich daraus:

- *Flexibilität bei der Unterstützung von Optionen*

Die IP-Header-Optionen können bei IPv6 wesentlich einfacher integriert werden. Durch einfache Erweiterungen des Header-Formats können neue Optionen jederzeit eingeführt werden.

- *Quality of Service*

Durch neue Steuerungsfelder im Header können die Datagramme zwischen dem Sender und dem Empfänger

anhand bestimmter Qualitätsparameter gesondert übermittelt werden. Dies wirkt sich besonders bei der Übermittlung von Real-Time-Anwendungen (Video und Sprache) aus.

- *Authentication und Privacy*

In der IP-Version 6 wurden die Bedürfnisse der heutigen Kommunikationsstrukturen berücksichtigt. Aus diesem Grund wurden Funktionen wie *Authentication*, *Datenintegrität* und *Sicherheit* implementiert.

- *Namensdienst*

Zu den weiteren Ergänzungen bei IPv6 zählt der IP-Namensdienst (DNS) in Form eines Sicherheits-Frameworks inklusive Authentisierung sowie Funktionen zur automatischen Konfiguration (Plug and Play, DHCP).

HINWEIS

Obwohl in der Version IPv6 die Adressfelder viermal länger sind als bei der alten Version, ist der gesamte Header nur doppelt so lang wie bei IPv4.

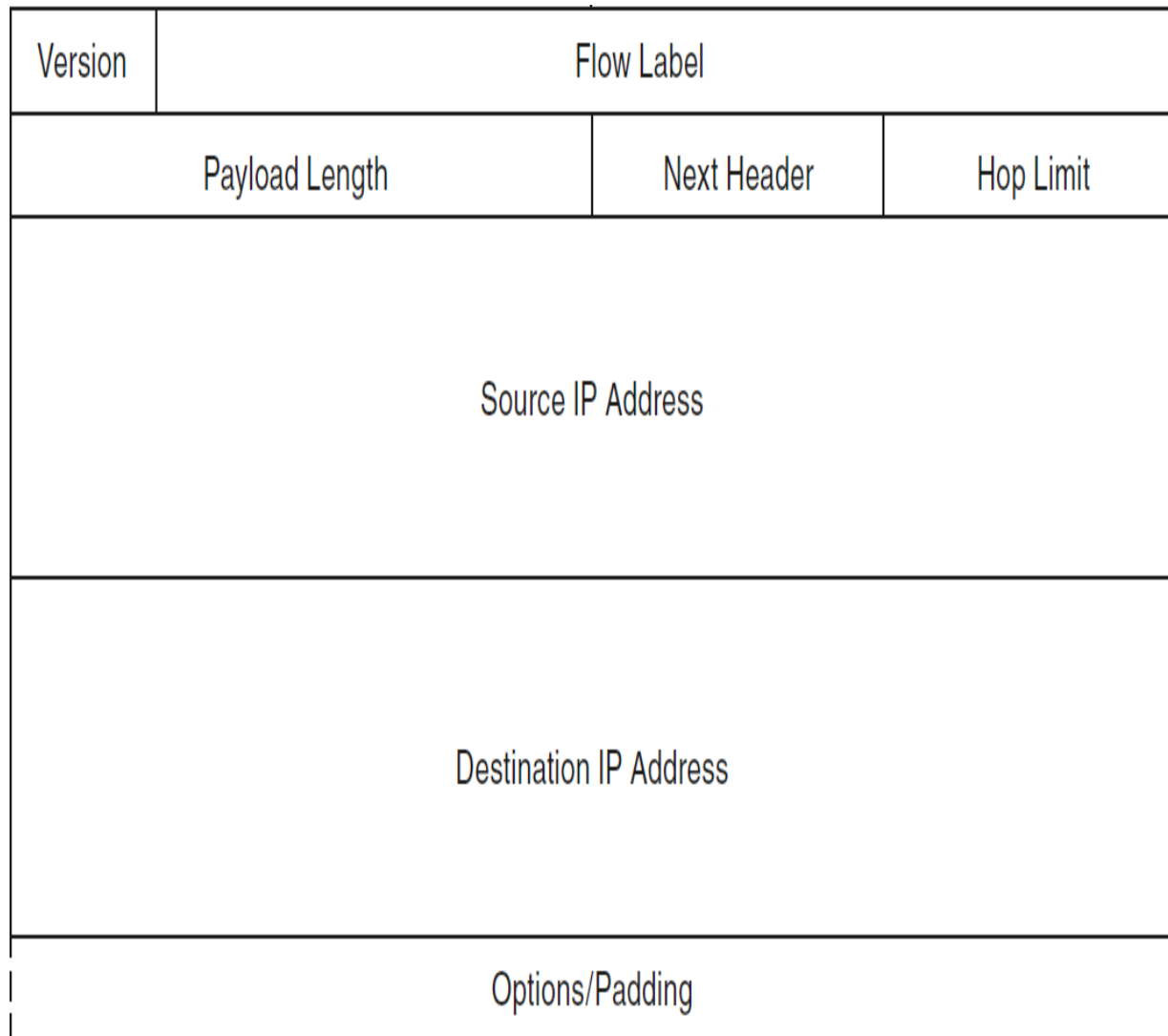


Abb. 3–10 *IP-Header der Version 6*

Die einzelnen Felder im IP-Header von IPv6 haben folgende Bedeutung:

- *Version* (4)
Hier wird die IP-Version eingetragen, also 6.
- *Flow Label* (28)
Hier wird der »Quality of Service« (QoS) definiert.

- *Payload Length* (16)

In diesem Feld wird die Länge der Nutzdaten eingetragen (ausschließlich der Länge des Headers).

- *Next Header* (8)

Hier wird der Header-Typ eingetragen, der dem IP-Header unmittelbar folgt. Es kann sich dabei z.B. um einen TCP-Header handeln. Es werden die gleichen Protokollwerte wie im TOS-Feld der IP-Version 4 benutzt.

- *Hop Limit* (8)

Dieses Feld entspricht dem TTL-Feld (*Time To Live*). Es kann maximal einen Wert zwischen 0 und 255 beinhalten. Jeder Router, den das IP-Paket passiert, reduziert den Wert um 1. Ist der Wert bei 0 angelangt, wird der Inhalt des Pakets verworfen, um die Netzbelastung durch endlos kreisende IP-Pakete zu reduzieren.

- *Source Address* (128)

Quelladresse bzw. Sender

- *Destination Address* (128)

Zieladresse bzw. Empfänger

- *Options* (n x 8)

Dieses Feld kann zwar beliebig groß sein, muss allerdings ein Vielfaches von 8 betragen. In diesem Feld werden Informationen hinterlegt über:

- Routing

- Fragmentierung
- Authentifizierung
- Security Encapsulation
- Hop-by-Hop Options
- End-to-End Options

Stand der Einführung von IPv6

Mittlerweile sind viele Betriebssysteme und Netzwerkkomponenten (Router, Switches usw.) verfügbar, die IPv6 unterstützen. Alle Tests der Vergangenheit führten zu relativ eindeutigen Aussagen, dass die entwickelte Software sehr gut läuft und beispielsweise auf Systemen parallel zu einer IPv4-Software eingesetzt werden kann. Es können weltweit IPv6-Adressen beantragt werden, wobei auch in Deutschland einige Internet-Service-Provider IPv6-Adressen beantragt und registriert haben.

Die Einführung eines neuen Adressierungsschemas lässt sich aufgrund der ständig notwendigen Verfügbarkeit des Internets nicht von einem auf den anderen Tag durchführen. Es muss gewährleistet sein, dass beide Systemvarianten (IPv4, IPv6) für eine gewisse Zeit nebeneinander eingesetzt werden können, ohne dass es dadurch zu Adresskonflikten o.Ä. kommt. Auch dies wurde bei der Entwicklung von IPv6 berücksichtigt, denn

IPv6 ist vollkommen kompatibel zu IPv4, was die Auswahl der Adressen angeht. So können mit IPv6 auch IPv4-Adressen angewählt werden, indem diese in IPv6-Adressen verpackt (eingekapselt) werden.

Im Rahmen des angedachten Migrationspfads mit einem Wechsel von Version 4 zu einem gemischten IPv4/IPv6-Betrieb und dem anschließenden Schritt zu einem reinen IPv6-Betrieb wurden von den Entwicklern drei Mechanismen vorgeschlagen, die unter der Bezeichnung *The simple IP Version Six Transition* (SIT) veröffentlicht wurden:

- IPv4-kompatible Adressen
- IPv6-in-IPv4 Encapsulation
- IPv6/IPv4 Header-Translation (Übersetzung)

Für die Unterstützung der Version-4-Adressen wird der heutige IP-Adressraum in den von IPv6 eingebettet. Für die Codierung der alten Adressen werden die ersten 12 Bytes der 16 Bytes langen neuen Adressen mit festen Nullen gefüllt. Die restlichen vier Bytes sind für die IPv4-Adressen reserviert.

Eine IPv4-Adresse (xxx.xxx.xxx.xxx) hat also unter IPv6 folgendes Aussehen: 0:0:0:0:0:0:xxx.xxx.xxx.xxx oder kurz ::xxx.xxx.xxx.xxx. Diese Adressen tragen bei der Version 6 die

Bezeichnung *IPv4-Only-Adressen*. Bezogen auf eine IPv4-Adresse der Form 192.168.1.14 sähe eine (eingekapselte) Adresse in IPv6-Notation wie folgt aus:

```
0000:0000:0000:0000:0000:0000:192.168.1.14
```

oder kürzer

```
::192.168.1.14
```

Sollte das angesprochene Gerät mit der IPv4-Adresse den IPv6-Standard nicht unterstützen, werden die ersten fünf IPv6-Quads auf 0 gesetzt. Dem sechsten Quad wird dann zur Kennzeichnung der Inkompatibilität das Byte FFFF zugewiesen. Die beiden restlichen Quads enthalten wieder die IPv4-Adresse. Dies würde sich wie folgt darstellen:

```
0000:0000:0000:0000:0000:FFFF:192.168.1.14
```

oder kürzer

```
::FFFF:192.168.1.14
```

Mit Einsatz der dritten Stufe der Umwandlung (*IPv6/IPv4 Header-Translation*) wird erreicht, dass über einen geeigneten Übersetzer (Translator) auch solche Netzwerke oder Systeme gekoppelt werden können, die nur eine der jeweiligen Protokollvarianten unterstützen. Diese Strategie ist aber lediglich in Umgebungen notwendig, wo Netze verbunden werden, die nur über die jeweils andere Variante verfügen.

NAT, CIDR und RSIP als Alternativen

Jeder Systemverantwortliche sollte sich darüber im Klaren sein, dass die Probleme mit fehlenden IP-Adressen nicht weniger werden, solange noch das betagte IPv4-Protokoll zum Einsatz kommt. Erschwert wird dabei der Umstieg durch ständige »Verschlimmbesserungen« an der IP-Version 4. So wird bereits seit Jahren versucht, Schwachstellen durch Weiterentwicklungen oder Zusätze zu beheben. Stellvertretend sollen hier Verfahren wie CIDR, VLSM und NAT genannt werden. Das Adressierungsproblem war lange vor der

Fertigstellung von IPv6 und vergleichbaren Produkten bekannt. Um entsprechende Adressprobleme zu umgehen, wurden mehrere Zwischenlösungen geschaffen, die teilweise heute noch Bestand haben. So entstanden insbesondere mit NAT (*Network Address Translation*) und mit CIDR (*Classless Inter Domain Routing*) zwei Lösungen, die das Problem der zu knappen Adressräume partiell und temporär lösen können.

Network Address Translation (NAT)

Mithilfe des *NAT* (*Network Address Translation*) kann das Adressproblem durch den Einsatz reservierter IP-Adressbereiche innerhalb eines Unternehmens oder einer Organisation entschärft werden. Die verwendeten Adressen brauchen nicht offiziell beantragt zu werden und können auch in unterschiedlichen Unternehmen oder Organisationen parallel eingesetzt werden. Um ein solches Netzwerk mit dem Internet zu verbinden, wandelt NAT die intern verwendeten (reservierten) IP-Adressen in offiziell gültige Adressen um und ermöglicht somit eine Kommunikation des eigenen Netzes mit dem Internet.

HINWEIS

Wenn ein Unternehmen oder eine Organisation private IP-Adressen verwendet, muss beim Kontakt mit dem Internet bzw.

anderen öffentlichen Netzwerken eine Adressumwandlung erfolgen.

Der NAT-Einsatz und die Benutzung eines reservierten (privaten) Adressbereichs macht es beispielsweise auch wesentlich einfacher, ohne viel Konfigurationsaufwand den ISP zu wechseln. Vorteilhaft ist ebenfalls, dass große Unternehmen und Organisationen einen geringeren Bedarf an offiziellen Adressen haben und daher nur einen kleinen Block weltweit eindeutiger IP-Adressen benötigen.

Der Nachteil des NAT-Verfahrens ist, dass beim Zugriff auf ein öffentliches Netz (Internet) ein *Network Address Translator* eingesetzt werden muss. Darüber hinaus ergeben sich auch Probleme, wenn Anwendungen eingesetzt werden, die auf eine direkte Ende-zu-Ende- oder Punkt-zu-Punkt-Verbindung zwischen den beteiligten Systemen aufsetzen, wo also das eine System die IP-Adresse des anderen Systems kennen muss. Dies ist bei NAT grundsätzlich nicht möglich, da NAT die IP-Adresse nur in der Verwaltungsinformation (IP-Header) umsetzt. Sobald das empfangende System die IP-Adresse aus dem Datenbereich ausliest, steht dort eine aus seiner Sicht falsche Adresse. Beim Ausfall des NAT-Anschlusspunkts offenbart sich ein anderer Nachteil, denn dann sind alle Systeme, die über ihn kommunizieren, nicht mehr erreichbar.

Classless Inter Domain Routing (CIDR)

Speziell wegen des Problems der schnell wachsenden Verwaltungsinformation in den Netzwerkknoten (Routing-Tabellen) wurde *CIDR* (*Classless Inter Domain Routing*) eingeführt. Mit CIDR können verschiedene Netze, für die im Normalfall in einer Wegleitungstabelle (Routing-Tabelle) je ein Eintrag benötigt würde, in einem einzigen Eintrag zusammengefasst werden. Damit werden die entsprechenden Tabellen kleiner und die Leistungsfähigkeit und Performance der Router wird verbessert.

CIDR ermöglicht, dass ein IP-Adressbereich in beliebige kleinere Teile aufgeteilt werden kann. Dabei kann die Aufteilung eines Adressblocks von der Internet-Registrierung über einen großen ISP, von dort über einen mittleren und kleinen ISP bis zum Netzwerk eines Unternehmens oder einer Organisation erfolgen. Die erfolgreiche Verwendung von CIDR ist an drei Voraussetzungen geknüpft:

- Routing-Protokolle müssen die Netzwerkpräfixinformation mit jeder Routing-Information mitschicken.
- Alle Router müssen einen konsistenten Weiterleitungsalgorithmus, basierend auf der längsten Übereinstimmung, verwenden.

- Damit die Zusammenfassung von Routen erfolgen kann, müssen die Adressen entsprechend der Netzwerktopologie verteilt werden.

Ein wesentlicher Vorteil von CIDR ist, dass es eine entscheidende Rolle bei der Kontrolle des Wachstums der Routing-Tabellen einnimmt. Um die Routing-Informationen zu reduzieren, ist es notwendig, dass das Internet in einzelne Adressierungs- oder Verwaltungsdomänen (*Administration Domains*) aufgeteilt wird. Innerhalb einer solchen Domain gibt es detaillierte Informationen über die Netze, die der Domain zugeordnet sind. Außerhalb der Domain ist nur das Netzwerkpräfix der Domain bekannt. Damit kann mithilfe eines einzigen Eintrags in der Routing-Tabelle eine Verbindung zu vielen Systemen hergestellt werden.

RSIP

Ein weiterer Versuch zur Lösung des IP-Adressproblems war die Entwicklung von *RSIP* (*Realm Specific Internet Protocol*). RSIP unterstützt eine direkte Punkt-zu-Punkt-Verbindung auf Basis der IP-Adresse. Dadurch sind Anwendungen wie Telekonferenzen, Videokonferenzen oder auch die IP-Telefonie (*Voice over IP*) mit RSIP möglich. Die Implementierung von RSIP ist jedoch sehr aufwendig, denn es muss sowohl in jedem

teilnehmenden System als auch in der Software der Netzknoten (Router) eingebunden werden. Die jeweiligen Anwendungen, die auf RSIP aufsetzen sollen, müssen zudem separat angepasst werden.

Fazit

Die mittlerweile weit verbreitete Einführung von NAT und CIDR hat den Ruf nach einer neuen IP-Version (IPv6) zunächst verstummen lassen. Die Standardisierung von IPv6 verschwand aus dem Blickfeld der Öffentlichkeit, man konzentrierte sich im Internet auf den Betrieb und die Optimierung von IPv4. Nachdem die Probleme mit NAT und CIDR mittlerweile unübersehbar sind, ist es an der Zeit, sich wieder langfristig auszurichten und IPv6 ins Auge zu fassen.

Für große Unternehmen und Organisationen mit entsprechender Netzwerkinfrastruktur, die unter Umständen auch noch weltweit operieren, ist es sinnvoll, sich frühzeitig in einer speziellen Testumgebung mit IPv6 zu befassen, internes Fachwissen aufzubauen und sich womöglich auch schon um die Zuteilung von IPv6-Adressen zu kümmern. In dem Zusammenhang ist jedoch immer zu bedenken, dass beim Übergang zu IPv6 eine ganze Reihe anderer Protokolle, die im weitesten Sinn zur TCP/IP-Protokollfamilie gehören, ebenfalls

angepasst werden müssen. Dazu gehören Routing-Protokolle wie OSPF (*Open Shortest Path First*), EIGRP (*Enhanced Interior Gateway Protocol*) oder auch der Namensdienst DNS (*Domain Name System*), der mit IPv6 in der Lage sein muss, sowohl 32-Bit- als auch 128-Bit-IP-Adressen zu verwalten.

Auch wenn sich die Hersteller unterschiedlicher Produkte bereits an der Version 6 ausrichten, ist IPv6 heutzutage noch nicht flächendeckend eingeführt und wird nicht selten immer noch mit Skepsis betrachtet. Dabei stehen die Konzepte für die erforderliche Übergangsperiode von Version 4 auf Version 6 als zentraler Punkt im Raum. Für den möglichst reibungsfreien Übergang und ein problem-loses Zusammenarbeiten der beiden IP-Versionen soll ein inkrementeller Übergang von IPv4-Hosts und IPv4-Routern nach IPv6 ebenso möglich sein wie der Übergang in einem Schritt.

HINWEIS

IPv6 wird IPv4 nicht zu einem Stichtag ablösen, sondern es wird eine lange Phase der Konvergenz bzw. Migration geben müssen.

Speziell die Übergangs- oder Migrationsphase stellt für viele Anwender die größte Hürde auf dem Weg zur neuen IP-Version

dar. Des Weiteren hängt die rasche Verbreitung der neuen Version natürlich auch von leistungsfähigen und stabilen Beispiel-Implementierungen ab. Aus diesen Gründen ist es selbst viele Jahre nach Festlegung der einzelnen Definitionen immer noch schwer abzuschätzen, ob und wie schnell sich IPv6 am Markt durchsetzen wird.

Neben der technischen Realisierung bzw. Umsetzung werden natürlich auch die Kosten und der Nutzen einer solchen Umstellung weitere entscheidende Faktoren sein. Hilfreich ist hier, dass viele der am Markt erhältlichen TCP/IP-Implementierungen mehr oder weniger auf die frei verfügbaren Quelltexte des BSD-UNIX zurückgehen. Daraus sollte man ableiten, dass es für viele Hersteller ohne große Probleme möglich sein wird, ihre Produkte anzupassen und damit für eine schnelle Integration von IPv6 zu sorgen.

Mit der Verfügbarkeit und den Möglichkeiten der Umsetzung bzw. Migration nach IPv6 wurden die Voraussetzungen für ein langfristiges Wachstum des Internetmarkts geschaffen. Dabei ist auch keinerlei Eile geboten, denn die Migration hin zur reinen IPv6-Umgebung kann nach und nach vollzogen werden. Durch strategische Überlegungen kann der Übergang zu einer wesentlich verbesserten IP-Version erheblich erleichtert werden. Die daraus resultierende Veränderung der TCP/IP-

Protokollfamilie kann der Systemverwalter über einen langen Zeitraum umsetzen.

HINWEIS

Der Einsatz von IPv6 lohnt sich nicht nur wegen der Beseitigung der Adressproblematik, sondern auch aus Gründen der Implementierung neuer Funktionen, die Schwachstellen beseitigen und sich aus den aktuellen Übertragungsformen automatisch ergeben.

Routing

Freitagnachmittag, zur Feierabendzeit, auf einem Großstadtbahnhof: Zahlreiche Züge fahren in den Bahnhof ein. Einige Güterzüge durchfahren den Bahnhof, einige Personenzüge halten an. Fahrgäste steigen aus, andere wiederum steigen in die Züge ein und fahren damit weiter. Auf mehreren Gleisen können zur gleichen Zeit unterschiedliche Züge abgefertigt werden. Zur Koordinierung dieses täglich wiederkehrenden Chaos werden ausgeklügelte und durchdachte Fahrpläne entwickelt und Wege festgelegt, die von den einzelnen Güter- und Personenzügen genau eingehalten werden müssen. Dabei lassen sich denkbare Alternativen in der Wegewahl durch eine kombinierbare Anfahrt verschiedener Bahnhöfe und die Benutzung mehrerer Weichensysteme erreichen. Diese vorhandene Infrastruktur muss ggf. verändert oder erweitert werden, wenn sich der Bedarf von weiteren Anbindungspunkten ergibt.

Ein solches Szenario lässt sich durchaus als gedankliche Grundlage für die Auseinandersetzung mit dem Thema *Routing im Netzwerk* verwenden, wobei die Züge hier durch Datenpakete (Datagramme) ersetzt werden. Bevor der eigentliche Transportvorgang relevant wird, muss genau über die Wahl des einzuschlagenden Weges nachgedacht werden.

Die Wahl des optimalen Weges ist eine der Hauptaufgaben des *Routerings*. An allen Verbindungsknoten, wo eine Änderung der Route vorgenommen werden kann, werden im Netzwerk »Bahnhöfe«, die sogenannten *Router* eingesetzt. Dabei kann es sich um Router handeln, die lediglich ein einziges Protokoll zu verarbeiten haben, oder auch um *Multiprotokoll-Router* (mehrere »Gleise«), die in der Lage sind, mehrere Protokolle gleichzeitig zu routen. Der Begriff »Weiche« ist eher mit einer Bridge bzw. Brücke zu vergleichen, da hier zur Wegesteuerung i.d.R. weitaus weniger Intelligenz erforderlich ist als bei einem Router. Filtermechanismen bei Routern sorgen dafür, dass nur die gewünschten Datenpakete passieren dürfen. Unerwünschte Datenpakete (»Schwarzfahrer«) werden herausgefiltert (bzw. »am Bahnhof hinausgesetzt«).

Grundlagen

Aufgaben und Funktion

Ein Router oder Internet-Router hat folgende Aufgaben zu erfüllen:

- Anpassung an spezielle Internetprotokolle, einschließlich IP, ICMP oder andere
- Verbindung zweier oder mehrerer paketorientierter Netzwerke

Ein Router muss in seiner Funktionsweise den Anforderungen eines jeden Netzwerks Rechnung tragen, wobei folgende Aufgaben entscheidend sind:

- Ein- und Auspacken (*Encapsulation, Decapsulation*) von Daten-Frames der jeweiligen Netzwerke
- Versand und Empfang von IP-Datagrammen bis zu der maximal möglichen Größe. Diese wird durch die MTU (*Maximum Transfer Unit*) repräsentiert
- Umsetzung der IP-Zieladresse in die entsprechende MAC-Adresse des jeweiligen Netzwerktyps
- Reaktion auf Datenflusssteuerung und Fehlerbedingungen, sofern vorhanden

Ein IP-Router empfängt und versendet IP-Datagramme und benutzt dabei wichtige Mechanismen wie beispielsweise *Speichermanagement* oder *Überlastkontrolle*. Er erkennt Fehlermeldungen und reagiert darauf mit der Erzeugung geeigneter ICMP-Meldungen. Wenn der TTL-Timer (*Time To Live*) eines Datenpakets abgelaufen ist, so muss das Paket aus dem Netzwerk entfernt werden, um unendlich zirkulierende Daten-Frames im Netzwerk zu vermeiden.

Eine weitere sehr wichtige Funktionalität stellt die Fähigkeit dar, Datagramme zu fragmentieren. Sehr große Pakete werden

in mehrere, der MTU-Größe entsprechende Teilpakete unterteilt und später wieder zusammengesetzt. Gemäß den ihm in der Routing-Datenbasis vorliegenden Informationen bestimmt der Router den nächsten Ziel-Hop für das zu transportierende Datenpaket.

Über bestimmte Formalismen bzw. unter Verwendung des IGP (*Interior Gateway Protocol*) werden Routing-Informationen zwischen den Routern eines Netzwerks ausgetauscht. Für die Kommunikation mit anderen Netzwerken werden Mechanismen innerhalb von EGPs (*External Gateway Protocol*) eingesetzt (z.B. Austausch von Topologie-Informationen). Für ein umfassendes Netzwerk- und eigenes Systemmanagement stehen leistungsfähige Funktionen zur Verfügung, wie beispielsweise Debugging, Tracing, Logging oder Monitoring.

Anforderungen

Es bestehen besondere Anforderungen an Router-Systeme, die zunehmend für die Verbindung von LANs über zum Teil weit gestreute WAN-Konstruktionen verantwortlich sind (*Global Interconnect Systems*). Router benötigen dynamische Routing-Algorithmen mit minimierter Prozessor- und Kommunikationslast. Von verschiedenen Router-Herstellern

werden sogenannte *Queuing-Modelle* angeboten, die eine optimierte Datenflusssteuerung zulassen.

Router unterstehen normalerweise keiner kontinuierlichen Beobachtung. Sie arbeiten als *Unattended Components*, wobei dafür gesorgt sein muss, dass ein Monitoring und Management dieser Geräte über das Netzwerk realisiert werden kann (z.B. TELNET-Sessions als Konsolenbetrieb zu Installations-, Konfigurations- und Managementzwecken).

Router müssen an die unterschiedlichsten Technologien für WAN- bzw. LAN-Zugriffsgeschwindigkeiten angepasst werden können. Der Betrieb einer relativ langsamen 64-kbit-ISDN-Festverbindung muss ebenso ermöglicht werden wie der Anschluss an ein 100-Mbit-Fast-Ethernet-Segment oder Gigabit-Netzwerke. Für den Einsatz in bereits bestehenden Rechnernetzen muss das Zusammenspiel der Router unterschiedlicher Hersteller (mit natürlich unterschiedlichen Betriebssystemen) ohne nennenswerte Probleme realisiert werden können. Nach Festlegung eines gemeinsamen Routing-Protokolls muss die Verständigung heterogener Router-Systeme untereinander einwandfrei arbeiten.

Ein Router hat die Aufgabe, jeden Header eines IP-Pakets genau zu überprüfen. Er tut dies, bevor irgendeine

inhaltsabhängige Aktion vorgenommen werden kann. Dadurch ist er in der Lage, fehlerhafte Pakete vor Weiterleitung an andere Netzwerkressourcen zu verwerfen. In diesem Zusammenhang spielt der TTL-Wert (*Time To Live*) eine große Rolle. Er gibt das aktuelle Alter eines Datagramms an und veranlasst den Router, in folgender Weise zu verfahren:

- Der TTL darf vom Router dann nicht überprüft werden, wenn dieser das entsprechende IP-Datagramm nicht weiterleiten muss.
- Ein Router darf kein Datagramm mit einem TTL-Wert von »0« erzeugen oder weiterleiten.
- Die Tatsache, dass ein TTL-Wert von »0« oder »1« vorliegt, darf einen Router nicht dazu veranlassen, das entsprechende Datagramm zu verwerfen. Wenn es an ihn selbst gerichtet ist oder andere relevante Gründe vorliegen, *muss* er den Versuch unternehmen, es zu empfangen.

HINWEIS

Die wesentliche Funktion des TTL-Felds in einem IP-Datagramm stellt die Vermeidung von endlos im Netzwerk kreisenden IP-Paketen bzw. die Beendigung von Internet-Routing-Schleifen dar. Aus der Praxis zeigt sich, dass der TTL-Wert bei einem großen Router-Netzwerk mit ca. 20 Routern

mindestens bei 40 liegen sollte; ein üblicher Default-Wert liegt bei 64.

Funktionsweise

Zur Verbindung mehrerer Netzwerke stellt das in [Kapitel 2](#) beschriebene *Bridging* oder *Switching* eine Alternative dar. Hier werden allerdings alle Transportaktivitäten größtenteils innerhalb des ISO/OSI-Layers 2 (Datensicherungsschicht) durchgeführt; Switching bildet in einigen Fällen eine Ausnahme (Layer-3- und Layer-4-Switching). Das Routing spielt sich hingegen immer auf Layer 3 (Netzwerkschicht) ab.

Dem Transport von IP-Datagrammen liegt ein bestimmter Algorithmus zugrunde, wobei für alle Formen der Weiterleitung von Datenpaketen (*Unicast*, *Multicast* und *Broadcast*) folgende Vorschriften gelten:

Der Router erhält das IP-Datagramm vom Layer 2 (*Data Link Layer*). Dabei erfolgt eine Auswertung des IP-Headers nach folgenden Gesichtspunkten:

- Die vom Link Layer angegebene Paketgröße muss für die Aufnahme des IP-Datagramms ausreichend dimensioniert sein. Es sind mindestens 20 Bytes erforderlich.
- Die IP-Checksumme muss korrekt sein.

- Das IP-Datagramm-Header-Feld muss für die Aufnahme des IP-Headers ausreichend dimensioniert sein. Dieser Wert umfasst die 20 Bytes Fix-Header und zusätzlich mögliche Optionsfelder.
- Das IP-Datagramm-Total-Length-Feld muss die Größe des *IP-Headers* samt IP-Daten umfassen.

Der Router vollzieht eine erste Paketbehandlung nach den im IP-Header angegebenen Optionen. Die Paketbehandlung für weitere Optionen wird zu einem späteren Zeitpunkt fortgesetzt, wobei der Router eine Auswertung der Ziel-IP-Adresse nach folgenden Kriterien vornimmt:

- Das IP-Datagramm ist für den Router selbst bestimmt und muss zu *Reassembly*-Zwecken zwischengespeichert werden.
- Das IP-Datagramm ist nicht für den Router bestimmt und muss zwecks Weiterleitung zwischengespeichert werden.
- Das IP-Datagramm muss zwischengespeichert werden, da es einerseits weitergeleitet werden muss und andererseits (eine Kopie) an den Router selbst gerichtet ist.

Unicast

Wenn ein Datenpaket durch einen Router an eine Unicast-Adresse weitergereicht werden soll, muss er den nächsten *IP-Address-Hop* bestimmen. Dabei überprüft der Router die

Zieladresse im Datenpaket und versucht zunächst unter Verwendung geeigneter Algorithmen zu ermitteln, ob er das Datenpaket direkt über seine Schnittstelle (*Interface*) im benachbarten (*adjacent*) Netzwerksegment zustellen kann oder ob ein weiterer Router mit dem Transport beauftragt werden muss. Die Wahl der zu verwendenden Netzwerkschnittstelle wird hier ebenfalls vorgenommen.

In einem nächsten Schritt wird überprüft, ob das Datagramm überhaupt weitergeleitet werden darf. Hierzu ist eine Analyse der Quell- und Zieladresse erforderlich. Handelt es sich bei der Zieladresse beispielsweise um eine Broadcast- oder Multicast-Adresse, so darf das Datenpaket nicht transportiert werden. Ähnliches gilt für Pakete mit Adressen, die über Paketfilter oder Access-Listen explizit nicht übertragen werden dürfen. Der Router vermindert nunmehr den TTL-Wert um mindestens 1 und überprüft, ob dieser den Wert 0 angenommen hat. Ist dies der Fall, so muss das IP-Datagramm verworfen werden.

Ein Teil der Paketbehandlung gemäß den im IP-Header angegebenen Optionen wurde bereits vor Festlegung der zuvor erläuterten Routing-Verfahren vorgenommen. An dieser Stelle werden nun die restlichen Optionen »verarbeitet«. Dann erfolgt die *IP-Fragmentierung*. Anschließend führt der Router die

Bestimmung der MAC-Adresse des nächsten IP-Hops durch, packt das IP-Datagramm in einen geeigneten LLC-Frame ein (z.B. gemäß IEEE 802.3) und stellt es in die Output-Queue (Zwischenspeicher für den Ausgang) der gewählten Netzwerkschnittstelle.

Multicast

Handelt es sich bei der IP-Zieladresse um eine Multicast-Adresse, so wird anhand der IP-Quell- und -Zieladressen, die dem Datagramm-Header entnommen sind, ermittelt, ob das Datagramm über die für die Weiterleitung vorgesehene Schnittstelle empfangen worden ist. Wenn nicht, wird das Datagramm stillschweigend verworfen. Die Methode zur Ermittlung der korrekten Schnittstelle für den Empfang hängt von den aktiven Multicast-Routing-Algorithmen ab. Eines der einfachsten Verfahren ist das RPF (*Reverse Path Forwarding*), bei dem die geeignete Empfangsschnittstelle dadurch ermittelt wird, dass man per Unicast für eine fiktive Übertragung vom Multicast-Empfänger zum eindeutigen Multicast-Versender die geeignete Schnittstelle zum Versenden festlegt.

Auf Basis der IP-Quell- und -Zieladressen aus dem Datagramm-Header ermittelt der Router die ausgehenden Schnittstellen des Datagramms. Um einen IP-Multicast für eine

ausgedehnte Ringsuche zu implementieren, wird für jede ausgehende Schnittstelle ein *Minimum-TTL-Wert* festgelegt. Auf jeder Schnittstelle (*Interface*), deren TTL-Wert kleiner oder gleich dem TTL-Wert des Datagramm-Headers ist, wird eine Kopie des Multicast-Datagramms versendet. Alle weiteren Schritte werden auf jeder Schnittstelle parallel ausgeführt; der Router reduziert den Paket-TTL-Wert um 1. Wie bei dem Unicast-Datagramm erfolgt anschließend die Fortsetzung der Verarbeitung aller restlichen Optionen, die Durchführung der IP-Fragmentierung, die Ermittlung der MAC-Adressen für den nächsten IP-Hop, die entsprechende IP-Encapsulation in den LLC-Frame und die Überstellung in den Zwischenspeicher des jeweiligen *Output-Interface*.

Broadcasts

Es gibt zwei Haupttypen von IP-Broadcast-Adressen: *Limited Broadcasts* und *Directed Broadcasts*. *Directed Broadcasts* werden in drei weitere Subtypen unterschieden: Broadcasts, die an ein spezifiziertes Netzwerkpräfix (Netz-ID) gerichtet sind, des Weiteren an ein Subnetz gerichtete Broadcasts sowie Broadcasts, die an alle Subnetze eines Netzwerks gerichtet sind. Die Klassifizierung eines Broadcasts hängt stets von seiner Adresse und der Kenntnis des Routers von der Struktur des

Zielsubnetzes ab. Von anderen Routern wird derselbe Broadcast möglicherweise anders interpretiert.

Die Wahl einer bestimmten Route, die ein Datenpaket auf seinem Weg durch das Netzwerk verwendet, ist von verschiedenen Kriterien abhängig. Die Bewertung dieser Kriterien und das Bestreben, eine Routenwahl zu optimieren, führen zu einem Routing-Verfahren, das sich durch seine Dynamik (im Routing-Algorithmus) vor allem bei Störungen im Netzwerk auszeichnet. Fest definierte Wege hingegen lassen nur ein geringes Maß an Flexibilität zu, können jedoch eine relativ gute Überschaubarkeit des Datenflusses gewährleisten.

Ein wichtiges Instrumentarium zur Bestimmung optimaler Routen ist die *Routing-Tabelle*, in der die zu verwaltende Netzwerktopologie abgebildet wird. Zwischen den im Netzwerk eingesetzten Routern werden diese Routing-Tabellen in bestimmten Zeitintervallen als wichtige Informationsquelle einander zugesandt und entsprechend ausgewertet.

Router-Architektur

Wie bereits erwähnt, findet das Routing auf dem Netzwerk-Layer statt. Das dabei verwendete Netzwerkprotokoll ist das *Internet Protocol* (IP), und zur Adressierung von IP-Knoten wird das in [Kapitel 3](#) dargestellte Adressierungsschema eingesetzt.

Network Layer

Gegenüber dem *Data Link Layer* (Schicht 2) verhalten sich Router auf Schicht 3 (*Network Layer*) transparent, d.h., die Zugriffsverfahren auf das physische Medium, wie Ethernet oder Token Ring, spielen im IP-Routing keine Rolle. Für beide Verfahren können daher das Internet Protocol und alle seine übergeordneten höheren Protokolle eingesetzt werden. So bereitet beispielsweise die Dateiübertragung mit dem IP-basierten Anwendungsprotokoll FTP keinerlei Probleme, wenn eine Empfangsstation am Ethernet-Netzwerk angeschlossen ist und die Sendestation in einem Token-Ring-Netzwerk betrieben wird.

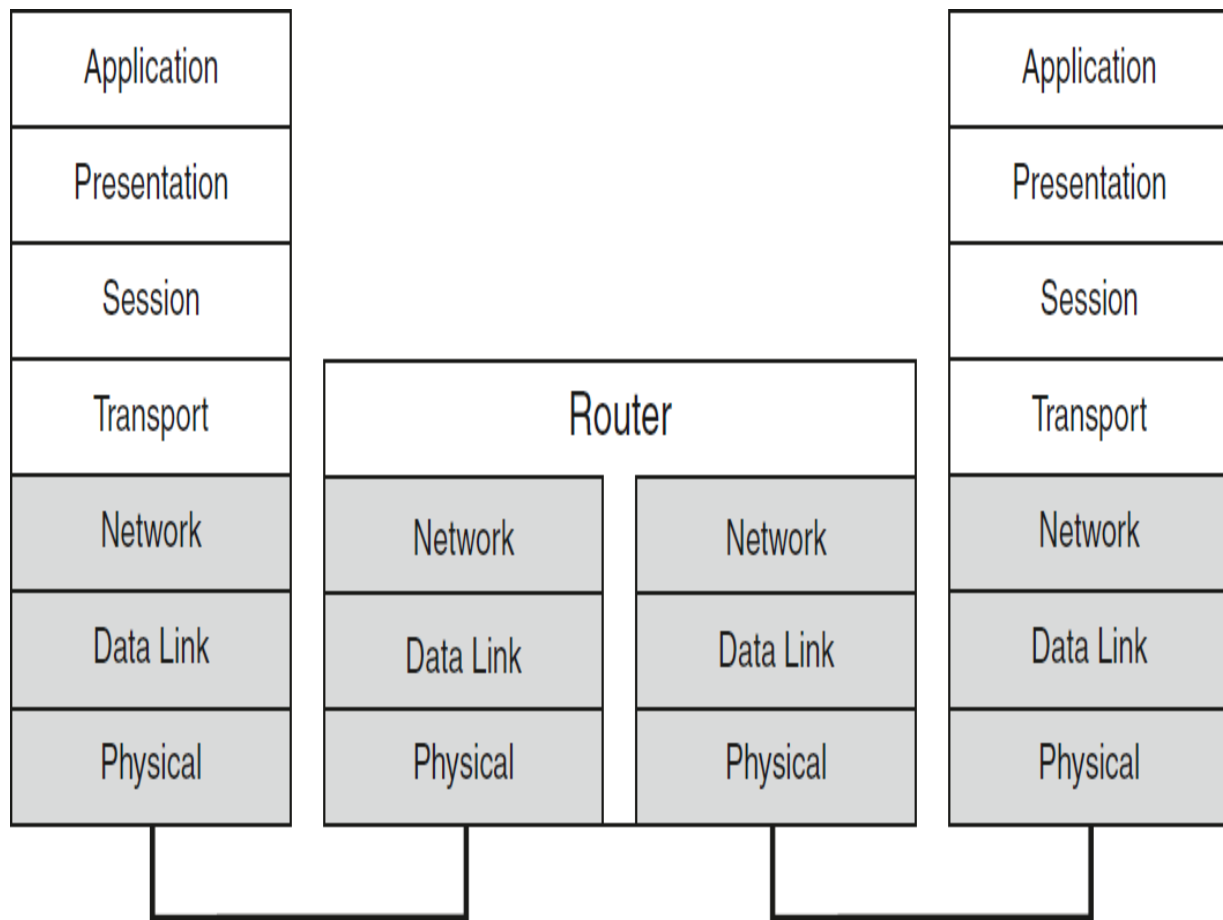


Abb. 4–1 Router-Einordnung im OSI-Modell auf Schicht 3

Direct Routing

Befindet sich die adressierte Zielstation im selben IP-Netzwerk, d.h., die Netz-ID der beiden Hosts ist identisch, werden die Dienste eines Routers nicht beansprucht, da eine direkte Adressierung vorgenommen werden kann, das sogenannte *Direct Routing*. Befinden sich zusätzlich Brücken in diesem Netzwerk, so haben diese auf das *Direct Routing* keinerlei Einfluss, da diese auf Layer 2 bekanntlich ein einziges logisches Netzwerk bilden.

Indirect Routing

Zur Adressierung eines Ziel-Hosts außerhalb des eigenen IP-Netzwerks (andere Netz-ID) wird mindestens ein Router-Übergang benötigt. Die Kommunikation wird also über einen Verbindungsknoten geführt, sie erfolgt nicht mehr direkt, sondern indirekt. Hier spricht man vom *Indirect Routing*. Die dazu erforderlichen Mechanismen (Routing-Verfahren, Routing-Tabellen, Routen usw.) werden in den folgenden Abschnitten näher beschrieben.

Default Routing

Das Prinzip des *Default Routing* wird immer dann verwendet, wenn es über die vorhandenen Einträge einer Routing-Tabelle nicht möglich ist, eine Zielstation zu erreichen oder aber noch keine Routing-Einträge (z.B. unmittelbar nach Installation eines Systems) existieren. Das Default Routing kann sowohl direktes als auch indirektes Routing verwenden. Der Default-Routing-Eintrag besteht normalerweise aus der fiktiven IP-Adresse 0.0.0.0 und der Adresse des zu benutzenden Routers.

Routing-Verfahren

Egal, welches Verfahren angewendet wird, die Grundlagen des Routings sind immer die sogenannten *Routing-Tabellen*. Sie

enthalten die für eine Erreichbarkeit verschiedener Netzwerke (unterschiedliche Netz-IDs) relevanten Informationen, wobei zwischen zwei Routing-Verfahren unterschieden wird: Das *statische Routing* arbeitet mit festgelegten Routing-Pfaden, die nicht verlassen werden dürfen. Das *dynamische Routing* hingegen basiert auf einer Kommunikation der Router untereinander und verfügt somit über das gesamte Topologiewissen des Netzwerks, wobei sich, je nach Anforderung, die Routing-Pfade ändern können.

Statisches Routing

Beim statischen Routing (siehe [Abb. 4-2](#)) werden die notwendigen Routing-Tabellen vom Verwalter des Netzwerks manuell angelegt und die entsprechenden Informationen in einer permanenten Datenbank hinterlegt. Erweiterungen oder Modifikationen müssen manuell in die Tabellen der jeweiligen Router eingetragen werden.

In dem Zusammenhang können Host-Routen oder *Netz-Routen* definiert werden. Ein *Host-Routing-Eintrag* beschreibt die Route zu einem ganz bestimmten Host in einem fremden Netzwerk. Ein anderer Host in diesem Netzwerk kann allerdings dann nicht erreicht werden. Es handelt sich quasi um eine Punkt-zu-Punkt-Verbindung innerhalb eines IP-

Netzwerkverbunds. Der Befehl zur Definition lautet beispielhaft:

```
route add host 190.137.23.76 190.136.10.1
```

Damit wird festgelegt, dass der Host 190.137.23.76 im Netzwerk 190.137.0.0 über den Router 190.136.10.1 aus dem Netzwerk 190.136.0.0 erreicht werden kann (als Subnetzmaske wird dabei 255.255.0.0 für Klasse-B-Netzwerke vorausgesetzt).

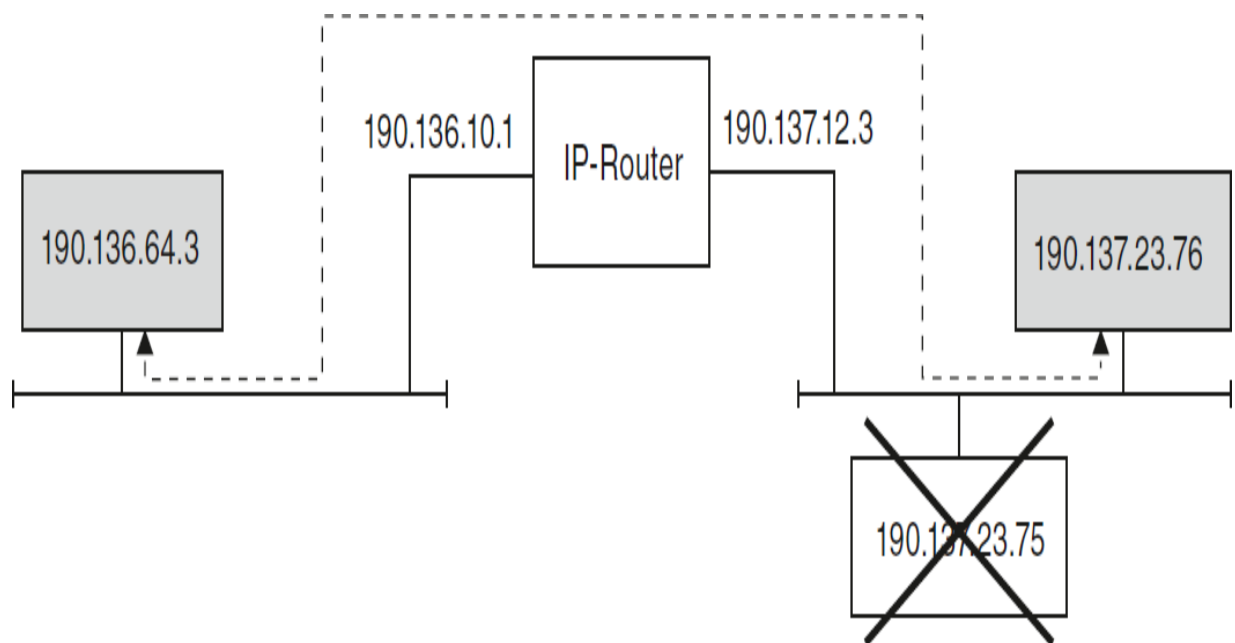


Abb. 4–2 Beispiel für statisches Routing

Die Alternative zu festen Einträgen für bestimmte Geräte (Hosts) lautet: *Netz-Routing-Einträge*. Dabei werden diejenigen

Router angegeben, die den Übergang in das fremde IP-Netzwerk topologisch ermöglichen. Es wird also nicht nur die Route zu einem bestimmten Host charakterisiert, sondern damit kann ein ganzes Netzwerk adressiert werden (Netzwerk mit einer anderen Netz-ID). Der Befehl zur Definition lautet beispielhaft:

```
route add net 190.137.0.0 190.136.10.1
```

Hier eröffnet der Router mit der IP-Adresse 190.136.10.1 dem Host des lokalen Netzwerks einen Zugang zu allen Hosts innerhalb des Netzwerks 190.137.0.0.

HINWEIS

Das statische Routing wird immer seltener eingesetzt, da die mittlerweile deutlich gewachsenen IP-Netzwerke es nicht mehr zulassen, manuelle Wartungsarbeiten an den Routing-Tabellen durchzuführen. Zudem ist der personelle Aufwand sehr hoch, sodass prinzipiell die Beschränkung auf sehr kleine Netzwerke bzw. für die Realisierung einer eingeschränkten Kommunikation aus Sicherheitsgründen praktikabel ist.

Dynamisches Routing

Das dynamische Routing (auch *Adaptive Routing* genannt) geht von einer umfangreichen Verständigung aller im Netzwerkverbund installierten Router untereinander aus. Um diese Router-Router-Kommunikation zu ermöglichen, werden spezielle Router-Protokolle eingesetzt. Die Dynamik dieses Routing-Verfahrens liegt in seiner Anpassungsfähigkeit gegenüber unvorhersehbaren Ereignissen im Netzwerk. Solche Ereignisse können Router-Ausfälle bzw. -Störungen sein, die bei statischen Routen dazu führen, dass Verbindungen abbrechen und bis zur Beseitigung der Störung nicht wieder etabliert werden können. Das dynamische Routing sorgt durch die eingesetzten Router-Protokolle, die im Grunde die »Intelligenz« der Router darstellen, und die Router-Router-Kommunikation dafür, dass Alternativwege ermittelt werden können und somit eine »Umleitung« um einen defekten Router herum gebildet werden kann. Von einem derartigen Ausfall merkt der Anwender in der Regel überhaupt nichts.

Darüber hinaus lassen sich im Netzwerk neue Hosts hinzunehmen, ohne dass manuelle Eingriffe im Router erforderlich werden. Die Protokollintelligenz erkennt neue Maschinen und fügt sie ihrer dynamisch erzeugten Routing-Tabelle hinzu. Einige Routing-Protokolle bieten zudem

Zusatzfunktionen wie Lastverteilung und ermöglichen somit eine dynamische Datenflusssteuerung je nach Datenverkehrsaufkommen.

Routing-Algorithmus

Folgende Objekte bzw. Mechanismen sind zur Beschreibung des Routing-Algorithmus erforderlich:

- *IP-Routing-Table*
Informationen zur Erreichbarkeit von IP-Hosts
- *Direct Routing (DR)*
Adressierung von Hosts im lokalen Netzwerk
- *Indirect Routing (IR)*
Adressierung von Hosts außerhalb des lokalen Netzwerks
- *Default Routing*
Adressierung von Hosts, die über DR und IR nicht erreicht werden können

Zur Veranschaulichung des Routings wird mit der folgenden Anweisung der Aufbau einer TELNET-Verbindung zu einem Zielrechner initiiert:

```
telnet 190.137.23.76
```

Die Zieladresse wird durch den Routing-Algorithmus erfasst und über die Routing-Tabelle geprüft (siehe [Abb. 4-3](#)). Dabei erfolgt zunächst die Anfrage, ob die Zieladresse im selben Netzwerk zu finden ist (DR). Ist dies nicht der Fall, so wird der Zielknoten im Routing-Eintrag für fremde Netzwerke gesucht. Lässt er sich dort ebenfalls nicht finden, so bleibt nur noch die Möglichkeit, den Default-Routing-Eintrag zu überprüfen. Enthält dieser eine Routenbeschreibung, die eine Kontaktaufnahme zum Zielrechner ermöglicht, so kann die TELNET-Session etabliert werden. Andernfalls erfolgt die Information des Anwenders durch Ausgabe der Fehlermeldung:

```
network unreachable - no route to host
```

Ziel-Netzwerkadresse	Routenwahl über
190.136.0.0	lokal
190.137.0.0	190.136.10.1
190.138.0.0	190.136.20.1

0.0.0.0 (default)	190.136.1.1
-------------------	-------------

Abb. 4–3 *Muster einer IP-Routing-Tabelle*

Durch Routing-Protokolle werden Router befähigt, ihre Routing-Tabellen mit anderen Routern auszutauschen bzw. Tabellenaktualisierungen (*Table Updates*) durchzuführen, wobei das Zeitintervall für die Updates konfigurierbar ist. Die auf diesem Wege ermittelten Informationen werden für die Ermittlung der günstigsten Route verwendet. Diese ändert sich ggf. so oft, wie ein Tabellen-Update erfolgt und eine Topologieänderung andere Routen ermöglicht. In dem Zusammenhang ist für die Bewertung einer optimierten Route die Anzahl von *Hops* (Netzübergänge durch Router) wichtig. In einigen Routing-Protokollen (z.B. *Open Shortest Path First* = OSPF) ist neben der Hop-Zahl auch noch eine Bewertung von Verbindungen durch fiktive Kostenwerte möglich. Damit wird die Routenwahl verfeinert und ist nicht nur von zwischengelagerten Routern abhängig. Bei LAN-LAN-, aber vor allem bei LAN-WAN-Verbindungen kann somit zwischen langsamen und schnellen Verbindungen differenziert werden.

Ein Router ist bestrebt, aus der Kenntnis seiner Routing-Informationen heraus die günstigste Route zu wählen (z.B. geringste Anzahl von Hops). Ferner sind Routing-Algorithmen

stets bemüht, positive Informationen, also optimierte Routen, schnell im Netzwerk über die Router-Router-Kommunikation (Routing-Tabellen-Updates) zu verbreiten. Negativinformationen – dazu gehören auch Routen, die durch Router-Störungen ausgefallen sind – werden jedoch nur schleppend im Netzwerk verbreitet. Dieser Umstand kann dazu führen, dass bei einem durch zahlreiche Router strukturierten Netzwerk die im Störfall notwendige Ermittlung der Alternativroute extrem lange dauert; eine Session-Unterbrechung bzw. lange Netzlaufzeiten sind die Folge.

Zur Gewährleistung der Kenntnis der aktuellen Netzwerktopologie mit allen möglichen Routen sind permanente Routing-Tabellen-Updates erforderlich, die jedoch aufgrund der damit verbundenen Netzlast nur in bestimmten Zeitintervallen vorgenommen werden können. Wenn also nach dem Ausfall eines Routers dieser keine Updates mehr versenden kann, sind die anderen Router stets auf die Hop-Informationen aus den alten Routing-Tabellen angewiesen. Diese sind allerdings nicht mehr aktuell, denn der defekte Router schickt schließlich keine Updates mehr. Die nun mit jedem Update immer älter werdenden Hop-Informationen führen zu einer kontinuierlichen Erhöhung des *Hop Counts*. Um eine mögliche Endlosschleife bei der Übermittlung über einen defekten Router zu vermeiden, wurde eine Hop-Anzahl von 15

als »nicht erreichbar« deklariert. Die maximale Anzahl von Routern zwischen zwei kommunizierenden IP-Hosts wurde daher auf 14 limitiert, wobei diese Problematik auch als *Slow Convergence Problem* bezeichnet wird.

Die Hop-Informationen als Entfernungsangabe zu den erreichbaren Netzwerken sehen im Normalfall für das in [Abbildung 4-4](#) dargestellte Netzwerk so aus wie in [Abbildung 4-5](#) dargestellt. Daraus geht beispielsweise hervor, dass eine Verbindung von dem Host 190.136.64.3 (*Router Münster* mit Netz-ID 190.136.0.0) zum Host 190.140.17.42 (*Router Ulm* mit Netz-ID 190.140.0.0) über insgesamt zwei Hops (Router) möglich ist.

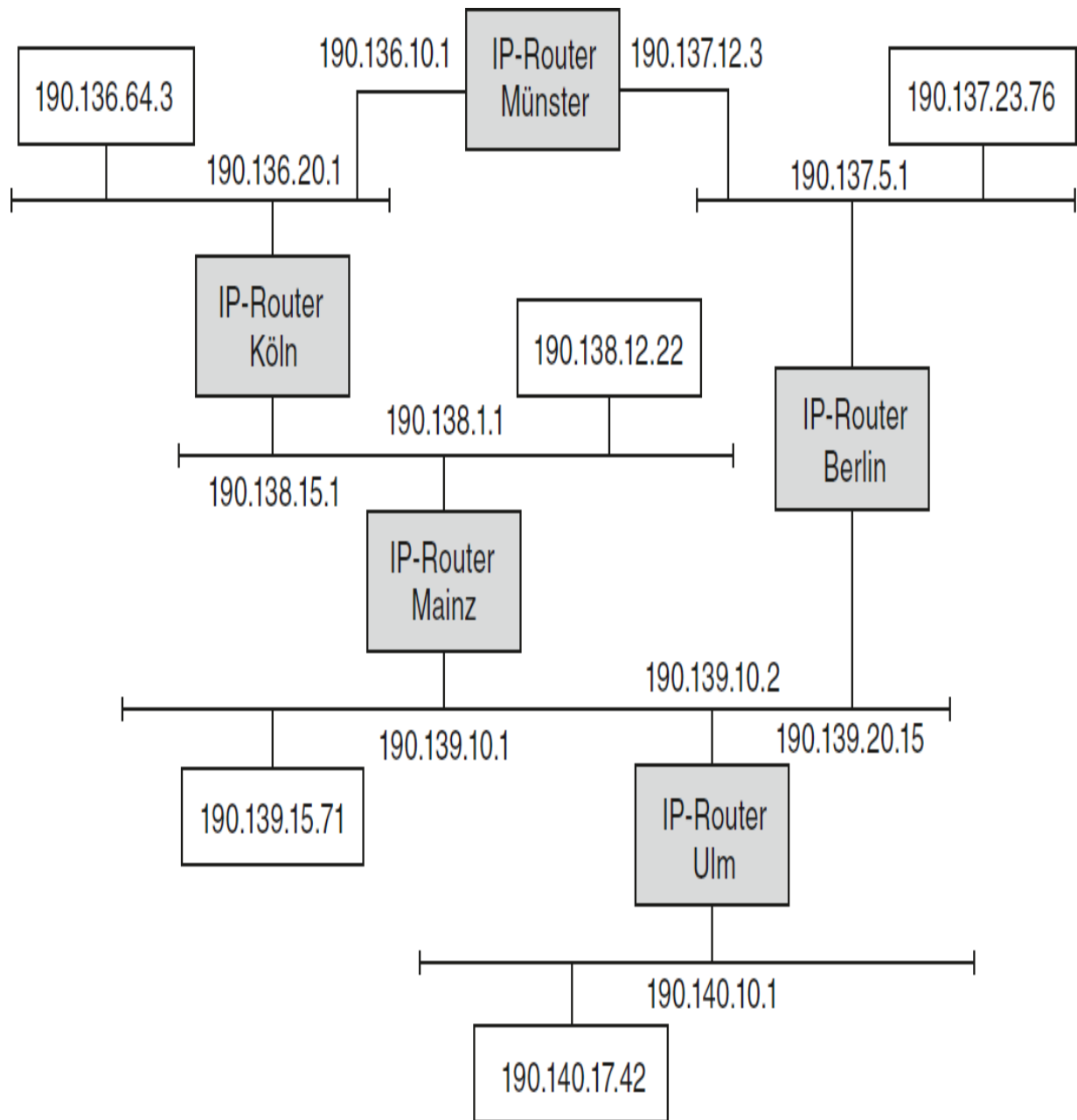


Abb. 4–4 *Beispiel für ein Netzwerk mit mehreren Routern (Hop-Count)*

Router Münster		Router Köln		Router Mainz		Router Berlin		Router Ulm	
190.136	0	190.136	0	190.136	1	190.136	1	190.136	2
190.137	0	190.137	1	190.137	1	190.137	0	190.137	1
190.138	1	190.138	0	190.138	0	190.138	1	190.138	1
190.139	2	190.139	1	190.139	0	190.139	0	190.139	0
190.140	2	190.140	2	190.140	1	190.140	1	190.140	0

Abb. 4–5 *Tabelle mit Informationen zu den einzelnen Routern (Hops)*

Einsatzkriterien für Router

Nicht selten werden Diskussionen über den Einsatzzweck und Sinn von Routern gegenüber der (manchmal) preiswerteren Alternative einer Implementierung von Brücken bzw. Switches geführt. Beides sind Strukturelemente, die zur Gestaltung der Infrastruktur eines Netzwerks eine wichtige Rolle spielen können, die jedoch von ihren Funktionen her nicht vergleichbar sind. Beide leisten einen entscheidenden Beitrag, wenn es um die Steuerung des Datenflusses geht, wobei einige Argumente für den Einsatz von Routern sprechen.

Fragmentierung

Werden zwei gleichartige Netzwerke über Brücken/Switches miteinander verbunden, so sind in der Regel keine Probleme hinsichtlich der Größe der übertragenen Daten-Frames zu erwarten. Handelt es sich bei dem ersten Segment jedoch um ein Ethernet-Netzwerk und bei dem zweiten Segment um ein Token-Ring-Netzwerk, so können sehr schnell Schwierigkeiten auftreten, wenn eine Station im Token Ring einen Frame aussendet, der eine erlaubte Größe von beispielsweise 4 KByte besitzt. Die MTU im Ethernet ist auf maximal 1500 Bytes beschränkt, sodass die verbindende Brücke den Token-Ring-Frame ablehnen muss bzw. ignoriert. Mit Routern ist diese Problematik zu vermeiden, denn sie beherrschen das Fragmentieren von Datenpaketen. Das Internet Protocol bzw. das übergeordnete TCP ermöglicht dem Router als Gerät der Kommunikationsschicht 3, ein Datenpaket zu stückeln und am Zielrouter wieder zusammenzusetzen. Die Gefahr einer Verwerfung von Datenpaketen, die wegen ihrer Größe nicht akzeptiert werden können, besteht beim Einsatz von Routern also nicht.

Error Handling

Durch die Verfügbarkeit eines leistungsfähigen Protokolls wie ICMP (*Internet Control Message Protocol*) lassen sich Fehlersituationen oder Störungen wesentlich besser im Router-Router- oder Router-Host-Datenverkehr übermitteln, und es kann schneller auf derartige Situationen reagiert werden.

Filtering

In den meisten Routern sind bereits umfangreiche Paketfilter implementiert, die den grundsätzlichen Sicherheitsanforderungen für eine einfache Abschottung nach außen genügen. Sie bieten damit auch die Grundlage für die Entwicklung hoch entwickelter Firewall-Systeme, die für eine Anbindung an das öffentliche Internet eine unverzichtbare Voraussetzung sind.

Die verfügbare Filterintelligenz lässt sich auch zur gezielten Protokollsteuerung einsetzen. Sollen nur ausgewählte Protokolle über einen Router geführt werden, um somit die Netzbelastung zu reduzieren, so kann dies durch eine geschickte Filterprogrammierung durchaus realisiert werden. Allerdings stellen umfangreiche und komplexe Filterdefinitionen hohe Anforderungen an die CPU-Leistung und sollten stets mit Bedacht formuliert werden. Für

Netzwerkumgebungen mit geringeren Anforderungen gibt es mittlerweile auch Low-Cost-Router, die recht preiswert sind, aber natürlich auch deutlich verminderte Performance-Werte liefern.

HINWEIS

Die Multifunktionalität von Routern und ihre Leistungsfähigkeit sind nur zu erreichen, wenn sie mit qualitativ hochwertiger Software, einem entsprechend großzügig dimensionierten Arbeitsspeicher und einer Hochgeschwindigkeits-CPU ausgestattet werden. Entsprechend hohe Investitionen sind fällig, wenn ein Router-Netzwerk aufgebaut werden soll.

Broadcast-Reduzierung

Die Entstehung unkontrollierter *Broadcasts* ist zumeist durch Protokollfehler oder Netzwerkstörungen bedingt. Sie werden durch Brücken nicht daran gehindert, das lokale Netzwerksegment zu verlassen und dadurch den gesamten Netzwerkverbund zu überfluten. Diese *Broadcast-Stürme* lassen sich im globalen Netzwerk durch den Einsatz von Routern vermeiden. Sie analysieren die Datenpakete und entscheiden nach Überprüfung, ob ein Broadcast weitergeleitet oder

verworfen werden soll. Dadurch lässt sich vermeidbarer Broadcast im lokalen Segment isolieren.

Insbesondere bei LAN-WAN-Verbindungen steht die Broadcast-Problematik an erster Stelle, denn »normale« Broadcast-Situationen oder gar Broadcast-Stürme, die als »Grundrauschen« im Hochgeschwindigkeits-LAN kaum wahrgenommen werden, machen sich im WAN sehr schnell negativ bemerkbar.

Routing-Protokolle

Die Intelligenz oder Logik des *Routings* ist in speziellen *Routing-Protokollen* implementiert; diese stellen auf verschiedene Art und Weise Verfahren zur Verfügung, um den Routing-Prozess zu realisieren. Je nach Bedarf lässt sich ein Protokoll wählen, das ein fest umschriebenes Spektrum an Funktionalität bietet. Oft hängt die Wahl des Routing-Protokolls auch von der eingesetzten Router-Hardware ab. Dies ist mittlerweile jedoch recht selten geworden, da fast alle Hersteller von Routern auch die wichtigsten Routing-Protokolle in ihren Geräten zur Verfügung stellen.

Neben den Routing-Protokollen gibt es zusätzlich die Bezeichnung der *routbaren* Protokolle. Darunter versteht man

ein Protokoll, das auf der Netzwerkschicht (Layer 3) adressiert werden kann. Ein Routing-Protokoll wird hingegen ausschließlich für die Kommunikation zwischen Routern verwendet. Nachfolgend sind die wesentlichen Varianten dieser beiden Protokolltypen aufgeführt:

Routbare Protokolle

- Internet Protocol (IP)
- Address Resolution Protocol (ARP)
- Reverse Address Resolution Protocol (RARP)
- Internet Control Message Protocol (ICMP)

Routing-Protokolle

- Border Gateway Protocol (BGP)
- Exterior Gateway Protocol (EGP)
- Open Shortest Path First (OSPF)
- Routing Information Protocol (RIP)
- DECnet Routing Protocol (DRP)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)
- IS-IS
- Integrated IS-IS

Router werden – wie andere Netzwerkkomponenten auch – über ihre MAC-Adresse angesprochen, wodurch diese wiederum über einen *ARP-Reply* bekannt gemacht werden. Handelt es sich um ein Datenpaket, das mittels Ethernet-Protokoll übertragen wird, wird in der Routing-Protokollsoftware derjenige Prozess aktiviert, der für eine Überprüfung des Ethernet-Pakets zuständig ist. Nach erfolgreicher Überprüfung wird der Ethernet-spezifische Teil des Pakets abgetrennt und die IP-Logik aktiviert. Diese überprüft ihrerseits den IP-Teil des Datenpakets. Ist auch diese Überprüfung erfolgreich (fehlerfrei), so können anschließend Mechanismen eingesetzt werden, die über *Access Control Lists* (ACL) das Datenpaket gezielt weiterleiten oder den Weitertransport verhindern. Die Routing-Tabelle liefert dabei die Information, welcher physische Port angesprochen und welcher Router in Anspruch genommen werden muss, um das Datenpaket an das jeweils adressierte Netzwerk (anhand der Netz-ID) zu transportieren. Handelt es sich bei dem nächsten Ziel-Netzwerk beispielsweise um ein Token-Ring-Netzwerk, so wird der vollständige Token-Ring-Frame zusammengebaut und über den entsprechenden Router-Port an das Netzwerk abgegeben.

Routing Information Protocol (RIP)

RIP ist ein schon recht »betagtes« Distance-Vector-Protokoll, das im Verlauf der Zeit einen überaus großen Verbreitungsgrad erfahren hat. Es wird trotz seiner teilweise nicht unerheblichen Mängel gern eingesetzt, nicht zuletzt deshalb, weil es nahezu auf allen Routern verfügbar und sehr leicht implementierbar ist.

Leistungsmerkmale

RIP (*Routing Information Protocol*) hat sich zu einem Zeitpunkt etabliert, als die Netzwerke noch relativ klein und überschaubar waren. Nur selten existierten innerhalb eines Netzwerks verschiedene Leitungsqualitäten mit unterschiedlichen Geschwindigkeiten. Diese Merkmale, die sich infolge der Heterogenität gewachsener Netzwerkstrukturen allmählich herausgebildet haben, stellen heute in den meisten Unternehmen eine Ausgangsbasis dar, die für ein homogenes Routing nicht unerhebliche Probleme aufwirft. Die Verfügbarkeit lediglich einer einzigen Metrik (Hop-Count) führt beim RIP oft zu einer nicht sehr realistischen Routenoptimierung, denn wenn allein die Anzahl der Hops bzw. der zu passierenden Router für eine Wegewahl entscheidend sein soll, so werden schnelle Netzabschnitte (z.B.

Fast Ethernet mit 100 Mbit/s) und deutlich langsamere (z.B. 64-kbit/s-ISDN-Festverbindungen im WAN) unter Umständen gleich gewichtet, was zu einer drastischen Fehleinschätzung der günstigsten Route führt. RIP lässt darüber hinaus die Bildung von Subnetzen nicht zu und schränkt somit seine Routing-Flexibilität deutlich ein.

In Zeitintervallen von 30 Sekunden erfolgt ein vollständiges Update der Routing-Tabellen. Geht man nun von einer Störung zwischen zwei Routern aus, so kann es bis zu 7,5 Minuten ($15 \text{ Hop-Counts} \times 30 \text{ Sekunden} = 7 \text{ Minuten } 30 \text{ Sekunden}$) dauern, bis RIP diese Störung erkennt und die betreffende Route aus seiner Tabelle streicht; denn erst wenn der nach jedem Tabellen-Update ermittelte neue Hop-Count für die Erreichbarkeit eines Ziels im Netzwerk den Wert 15 erreicht hat, wird die Aussage »nicht erreichbar« getroffen.

Bewertung

Eine Übersicht der Vor- und Nachteile lässt den Schluss zu, dass RIP nicht mehr zeitgemäß ist. Kleine und einfache Netzwerke jedoch können durch RIP-Funktionalität im Großen und Ganzen ausreichend bedient werden.

- Vorteile
 - sehr einfach zu implementieren

- nahezu überall verfügbar
- Algorithmus recht einfach und daher leicht zu durchschauen
- Public Domain
- Nachteile
 - keine Subnetzadressierung
 - lange Reaktionszeit bei Störungen (schlechte Konvergenz)
 - einzige Metrik ist der Hop-Count
 - unterschiedliche LAN/WAN-Geschwindigkeiten lassen sich nicht berücksichtigen
 - hoher LAN/WAN-Traffic durch vollständige Routing-Tabellen-Updates in festen Intervallen

Implementierung

Die Implementierung des RIP erfolgt unter Verwendung des sogenannten *routed* bzw. *route daemons*. Es handelt sich dabei um einen Prozess, der auf allen dedizierten Routern automatisch aktiviert ist und auf Workstations (UNIX-Maschinen oder auch Personalcomputern wie z.B. Windows-Server als Dienst) mit Router-Funktionalität und (mindestens) zwei Netzwerkcontrollern optional gestartet werden kann.

HINWEIS

In vielen Fällen wird beim RIP eine *Default Route* (meist 0.0.0.0) angegeben, mit der die Festlegung von Routen definiert wird, die dann zu benutzen sind, wenn die adressierte Netzwerkadresse vom Router nicht erreicht werden kann.

RIP-Version 2

Die Weiterentwicklung von RIP in der Version 2 stellt grundsätzlich kein völlig neues Routing-Protokoll dar, sondern liefert lediglich einige wichtige Erweiterungen zur alten Version 1. Details hierzu sind im RFC 2453 vom November 1998 nachzulesen. So gibt es einige geringfügige Unterschiede hinsichtlich des *Message-Formats*. Während der *Frame-Header* in beiden Versionen identisch ist, weist der *Frame-Body* Unterschiede auf.

Der RIP-Header besteht aus den Feldern:

- *command (8)*

Ein in diesem Feld eingetragener Wert von »1« markiert das Datagramm als RIP-Request, der Wert »2« bezeichnet eine RIP-Response.

- *version (8)*

Dient zur Identifikation der RIP-Version.

- *reserved (16)*

Dieses zwei Oktette umfassende Feld ist mit binären Nullen belegt.

In der RIP-Version 1 wird dem Header ein sogenannter *RIP entry* angefügt, der aus insgesamt 20 Oktetten besteht. Jeder *RIP entry* umfasst die folgenden Felder:

- *address family identifier (16)*

Besitzt den Wert »2« in der RIP-Version 1, in RIPv2 kann der AFI unterschiedliche Werte annehmen.

- *reserved (16)*

Dieses zwei Oktette umfassende Feld ist mit binären Nullen belegt.

- *IPv4 address (32)*

Ziel-IP-Adresse

- *reserved (32)*

Dieses vier Oktette umfassende Feld ist mit binären Nullen belegt.

- *reserved (32)*

Dieses vier Oktette umfassende Feld ist mit binären Nullen belegt.

- *metric (32)*

Anzahl der erforderlichen Hops (Metrik) bis zur Erreichung des Ziernetzwerks. Es sind Werte zwischen 1 und 15

definierbar; der Wert 16 wird als »Netzwerk unerreichbar« interpretiert.

Die in RIPv2 erstmals vorgesehene Authentifizierung belegt einen vollständigen (und zwar den ersten) 20-Bytes-Routeneintrag. Allerdings kommt hier lediglich eine einfache Kennwortauthentifizierung zur Anwendung.

- *Identification (16)*

Zwei Bytes mit dem Inhalt 0xFFFF für die Kennung

- *Authentication type (16)*

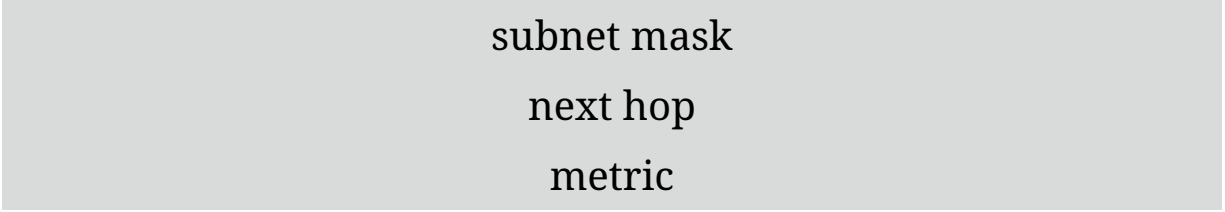
Dieses Feld gibt den Authentifizierungstyp an. Derzeit ist allerdings nur der Typ *Passwort* definiert.

- *Authentication (128)*

Dieses Feld enthält das Kennwort (maximal 16 Zeichen bzw. Bytes).

Die verbleibenden maximal 24 Routeneinträge können dann gemäß RIPv2-Message-Format (siehe [Abb. 4–6](#)) gebildet werden. Hier wird dann im Unterschied zur RIPv1 eine Subnetzmaske eingeführt.

command	version	reserved
address family identifier (afi)		route tag
IPv4 address		



A diagram showing a rectangular box divided into three horizontal sections. The top section is labeled 'subnet mask', the middle section is labeled 'next hop', and the bottom section is labeled 'metric'.

subnet mask
next hop
metric

Abb. 4–6 *RIPv2-Message-Format*

- *address family identifier (16)*
Siehe RIPv1-Frame
- *route tag (16)*
Kennzeichen zur Differenzierung von internen und externen RIP-Routen
- IPv4 address (32)
Siehe RIPv1-Frame
- *subnet mask (32)*
Subnetzmaske der IP-Adresse
- *next hop (32)*
IP-Adresse des nächsten Hops (innerhalb des lokalen Subnetzes)
- *metric (32)*
Siehe RIPv1-Frame

Open Shortest Path First (OSPF)

Als ein »Konkurrent« zum *Routing Information Protocol* hat sich *OSPF (Open Shortest Path First)* als modernes *Link State*

Protocol auf dem Routing-Markt etabliert. Die offiziellen Charakteristika des mittlerweile standardisierten Routing-Protokolls *OSPF Version 2* sind im RFC 2328 vom April 1998 nachzulesen. OSPF wird eigentlich als das Nachfolgeprotokoll des RIP betrachtet. Allerdings ist eine kontinuierliche Entwicklung vom RIP zum OSPF nicht zu erkennen, da beiden Protokollen recht unterschiedliche Philosophien zugrunde liegen. In der Praxis hat OSPF zwischenzeitlich das RIP als Standard für Routing-Protokolle abgelöst. OSPF wartet mit Funktionen auf, die RIP nicht zu bieten hat, als da beispielhaft zu nennen sind:

- Verwendung von Subnetzen und variablen Subnetzmasken
- Authentifizierung
- Einsatz mehrerer Metriken (Hop-Count, Kosten, Zuverlässigkeit)
- Lastverteilung über Routen mit gleicher Kostenbewertung
- Priorisierungsmechanismen über das TOS-Feld des IP
- Bildung von Routing-Tabellen durch Link-Informationen der Router-Nachbarn
- Verwendung *kurzer* Datagramme aus Gründen der Netz-Performance

Netzwerkstruktur

Die übergeordnete Struktur in einem OSPF-Router-Netzwerk ist das *Autonome System* (AS). Es wird normalerweise die gesamte Netzwerkstruktur eines Unternehmens im LAN- und im WAN-Bereich umfassen. Jeder involvierte Router besitzt stets aktuelle Informationen über die Topologie des AS, aus der er seine relevanten Routing-Pfade ermitteln kann. Zur besseren Verwaltung eines solch komplexen Netzwerks bzw. eines AS wird allerdings eine Unterteilung in mehrere *Areas* vorgenommen, die jeweils individuell adressiert werden müssen (siehe [Abb. 4–7](#)).

Die Identifikation der *Areas* erfolgt durch eindeutige IDs, wozu in der Regel die jeweilige IP-Adresse der Area verwendet wird. Über eine Datenbank erhält jede Area die Kontrolle über ihre eigenen Topologie-Informationen. Diese werden im Router verwaltet. Router, die zwischen zwei Areas installiert sind, administrieren daher auch zwei verschiedene Topologiedatenbanken, die kontinuierlich abgeglichen werden. Aufgrund dieser Unterteilung spricht man einerseits vom *Intra-Area Routing*, das sich auf Routing-Aktivitäten ausschließlich innerhalb einer Area bezieht, und andererseits vom *Inter-Area Routing*, das sich mit der notwendigen Kommunikation der einzelnen Areas untereinander beschäftigt. Die an den Area-

Grenzen befindlichen Inter-Area-Router besitzen einen Zugang zu der sogenannten *Backbone-Area*, die alle einzelnen Areas zu einem AS verbindet. Diese Area wird im OSPF mit der Identifikation 0.0.0.0 versehen.

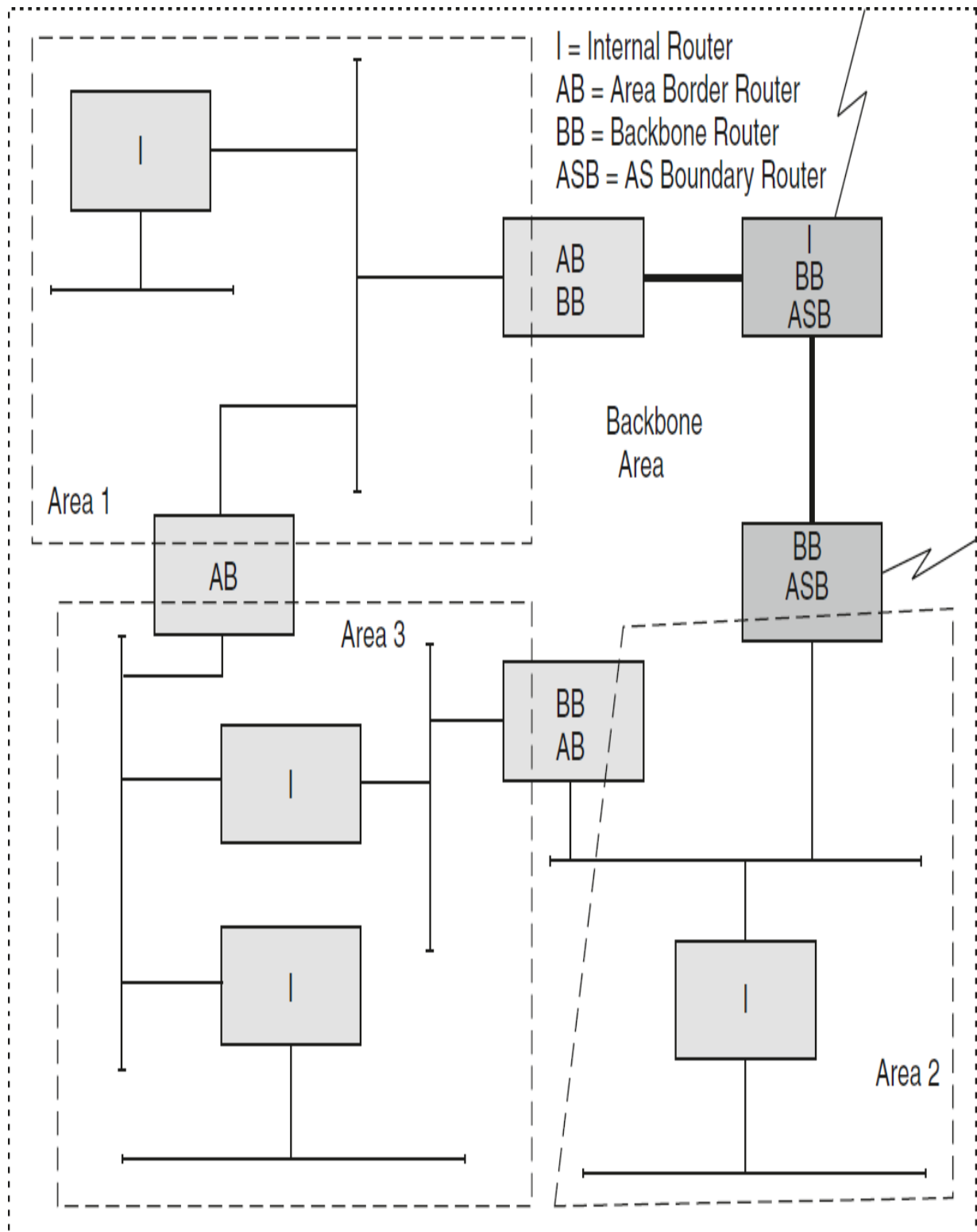


Abb. 4-7 Struktur der Areas im OSPF

Als wesentliche Bedingung für die Definition einer Area muss gewährleistet sein, dass jedes Netzwerk innerhalb einer Area erreicht werden kann, ohne diese dabei zu verlassen. Sollte dies aus bestimmten Gründen nicht eingehalten werden können (die Backbone-Area ist beispielsweise in sich nicht geschlossen, sondern zweigeteilt), so müssen »virtuelle Links« definiert werden, die eine zu konfigurierende logische Verbindung bilden, um die Backbone-Area wieder zu schließen. Für die Durchführung von Kommunikation innerhalb und außerhalb eines AS samt seiner Areas werden vier Router-Typen benötigt, die an bestimmten topologischen Schlüsselstellungen im Netzwerk positioniert werden:

- *Internal Router*

Befinden sich innerhalb einer Area.

- *Area Border Router*

Befinden sich an den geografischen Grenzen der Areas.

- *Backbone Router*

Befinden sich mit mindestens einer Schnittstelle in der *Backbone-Area*.

- *AS Boundary Router*

Befindet sich an der Grenze des AS – zur Verbindung mit weiteren AS.

Der Kommunikationsweg einer Sendestation zu einer Zielstation über Area-Grenzen hinweg lässt sich in folgende Phasen einteilen:

- *Phase 1*
Sendestation zum Area Border Router 1
- *Phase 2*
Area Border Router 1 zum Area Border Router 2 (Backbone-Weg)
- *Phase 3*
Area Border Router 2 zur Zielstation

Netzwerktypen

Folgende Netzwerktypen werden von OSPF unterstützt:

- *Point-to-Point Networks*
Dabei wird eine Verbindung zwischen zwei Routern über eine direkt angeschlossene Leitung realisiert, wobei es sich nicht ausschließlich um ein physisches Kabel handeln muss, das beide Router verknüpft; die Punkt-zu-Punkt-Verbindung kann auch über ein öffentliches Netzwerk wie beispielsweise über das ISDN (Festverbindungen, Wählverbindungen) hergestellt werden.
- *Broadcast Networks*

In einem *Broadcast Network* können mehrere Router eingebunden werden. Die Verständigung der einzelnen Netzknoten kann über *Broadcasts* erfolgen, wobei Mitteilungen gleichzeitig an alle (oder an eine Gruppe) Netzwerkteilnehmer verschickt werden.

- *Non-Broadcast Networks*

Dieser Netzwerktyp ermöglicht mehrere parallele Verbindungen zu verschiedenen Zielen, jedoch ist die Aussendung von *Broadcasts* nicht möglich. X.25-Netzwerke gehören zu diesem Typ, wobei virtuelle Verbindungen als SVC (*Switched Virtual Circuit*) oder PVC (*Permanent Virtual Circuit*) die Grundlage bilden.

HINWEIS

Die beiden zuletzt genannten Typen werden auch als *Multi-Access Networks* bezeichnet, da dabei in der Regel immer mehrere Router zum Einsatz kommen.

Arbeitsweise

Nach Aktivierung eines OSPF-Routers versucht dieser, über *HELLO-Messages* seine Router-Nachbarn zu erreichen. Dies geschieht in *Point-to-Point-Netzwerken* durch feste Adresszuordnungen, in *Multi-Access-Netzwerken* wird dazu die

Multicast-Adresse 224.0.0.5 verwendet. Eine solche Adresse der D-Klasse steht für die »normale« Adressierung von IP-Knoten nicht zur Verfügung. Sie dient lediglich dazu, Routing-Informationen an eine Gruppe von Routern zu senden. Dadurch kann die Verwendung von *Broadcasts* verhindert werden, um den Netzwerkverkehr erheblich zu reduzieren (nicht jeder Knoten erhält – wie bei einem Broadcast – die jeweilige Information, sondern lediglich die Router).

Im *Multi-Access-Netzwerk* wird daraufhin unter den involvierten Routern ein sogenannter *Designated Router* (DR) sowie sein Vertreter, ein *Backup Designated Router*, bestimmt (während dieser Zeit ist das gesamte Netzwerk blockiert). Der DR ist fortan für die gesamte Steuerung des Netzwerks zuständig. Bei einem DR-Ausfall übernimmt der Backup-DR die Rolle des DR. Die Kommunikation zwischen DR und Backup-DR erfolgt über die Multicast-Adresse 224.0.0.6. Sie wird ebenfalls von denjenigen Routern benutzt, die dem DR bzw. dem Backup-DR Informationen zukommen lassen wollen. Eine der Hauptaufgaben des DR ist die Bestimmung von *Adjacencies* (Nachbarschaften). Sie bezeichnen eine Gruppe von Routern, die direkt miteinander kommunizieren. Router in unterschiedlichen *Adjacencies* stehen normalerweise untereinander nicht in Kontakt. Sie kommunizieren lediglich mit den beiden DRs. Würde eine netzwerkweite

Kommunikation aller Router untereinander freigegeben, so führte dies zweifelsohne zu einer starken Netzwerkbelastung.

Router versenden in konfigurierbaren Zeitintervallen *Link State Advertisements* (LSA), in denen ihr aktueller Zustand beschrieben wird. Bleibt dieser Zustand konstant, so wird erst nach Ablauf des Zeitintervalls wieder ein LSA geschickt. Ändert sich jedoch sein Zustand, so erfolgt unmittelbar ein zusätzliches *Link-State-Update*. Aus der Vielzahl von LSAs ermittelt ein Router die Netzwerktopologie, für die er zuständig ist. Der *Shortest-Path-Algorithmus* berechnet aus diesen Informationen schließlich den günstigsten Weg.

In Abhängigkeit vom Router-Typ werden unterschiedliche LSA-Typen eingesetzt, um entsprechende Informationen zu versenden:

- *ROUTER LINK ADVERTISEMENT (type 1)*

In diesem LSA versendet ein normaler Router die Statusinformationen seiner Netzwerkschnittstelle innerhalb seiner eigenen Area.

- *NETWORK LINK ADVERTISEMENT (type 2)*

Innerhalb einer Area versendet der DR eine Aufstellung aller im Netzwerk installierten Router.

- *SUMMARY LINK ADVERTISEMENT (type 3)*

Diese LSA enthält Routenbeschreibungen und wird von Area-Border-Routern benutzt, um Ziele außerhalb der Area erreichen zu können.

- *SUMMARY LINK ADVERTISEMENT (type 4)*

Diese LSA enthält Routenbeschreibungen zu den AS-Boundary-Routern und wird von Area-Border-Routern benutzt, um Ziele außerhalb des Autonomen Systems (AS) erreichen zu können.

- *AS EXTERNAL LINK ADVERTISEMENT (type 5)*

Alle Router im AS bzw. in der Domäne werden vom AS-Boundary-Router informiert, wie ein Ziel innerhalb eines anderen AS erreicht werden kann.

Die unterschiedlichen Statuswerte, die der Netzwerkcontroller eines Routers annehmen kann, sind nachfolgend kurz beschrieben. Aus ihnen geht der aktuelle Zustand des Routers hervor:

- *Down*

Initialzustand eines Routers unmittelbar nach seinem Einschalten, jedoch noch vor der Initialisierung des Netzwerkcontrollers

- *Loopback*

Dieser Zustand verweist auf die Möglichkeit, bereits einfache Netzwerktests durchführen zu können (z.B. *Ping*). Normaler

Datenverkehr ist allerdings (noch) nicht möglich.

- *Waiting*

In diesem Zustand befindet sich ein Router bzw. seine Schnittstelle, wenn er das Netzwerk nach den bereits bekannten *HELLO-Messages* seines DR abhört. Erfolgt innerhalb eines bestimmten Intervalls (*RouterDeadInterval*) kein Empfang einer *HELLO-Message* vom DR, so muss ein neuer DR bestimmt werden.

- *Point-to-Point*

Der Netzwerkcontroller in einem *Point-to-Point-Netzwerk* oder an einem virtuellen Link befindet sich in einem aktiven Zustand. Es wird versucht, den Aufbau einer *Adjacency* mit dem Nachbar-Router vorzunehmen.

- *DR other*

Beim Router dieses aktiven Netzwerkcontrollers handelt es sich um keinen DR oder einen Backup-DR. Er baut jedoch *Adjacencies* zu ihnen auf.

- *Backup*

Beim Router dieses aktiven Netzwerkcontrollers handelt es sich um den Backup-DR.

- *DR*

Router dieses aktiven Netzwerkcontrollers ist ein *Designated Router*.

Topologiedatenbasis

Ein OSPF-Router erhält über *Link State Advertisements* (LSA) die Topologie-Informationen, die zur Bildung einer entsprechenden Routing- bzw. Topologiedatenbank in einem Router benötigt werden. Der Router selbst betrachtet sich bei Berechnung aller Routen in seiner Area als Root (Wurzel). Von ihm ausgehend werden alle potenziellen Wege über Router und Netzwerke »durchwandert«. Dabei werden Router-Netzwerkübergänge mit Kosten bewertet. Je geringer die Kosten, desto günstiger die Route. Eine solche primäre Route wird nur dann ignoriert, wenn eine Störung auf ihrem Pfad vorliegt. In dem Fall entscheidet sich OSPF für eine nach der Routenberechnung höher bewertete, also kostspieligere Routenalternative.

Abbildung 4–8 zeigt eine Topologie, die als Schaubild die Grundlage für die Bewertung der Routen darstellt. Der System-Manager hat dabei die Aufgabe, nach den ihm vorliegenden Kriterien (z.B. Leitungsqualität, Leitungs- bzw. LAN-Geschwindigkeit) eine kostenorientierte Bewertung einzelner Router-Netzwerkabschnitte vorzunehmen. Dabei wird ein Netzwerkabschnitt stets mit »0« bewertet. Alle übrigen Abschnitte werden mit Kostenwerten versehen. Eine schnelle LAN-Verbindung eines Routers zu einem Gigabit-Ethernet-Netzwerk könnte demnach mit einem geringeren Kostenwert

als eine langsame Ethernet-Verbindung mit hoher Kollisionsrate belegt werden.

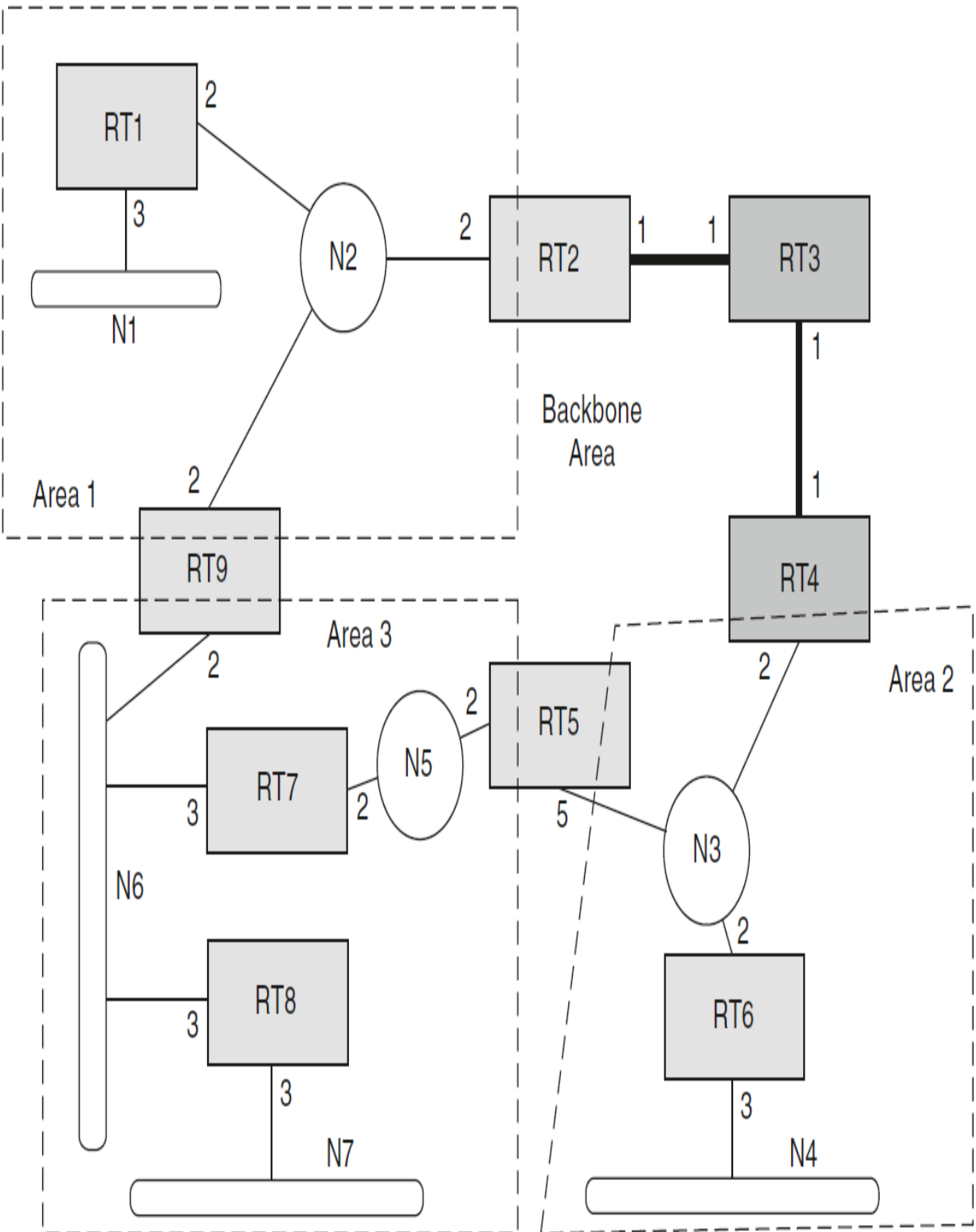


Abb. 4–8 OSPF-Topologie

Aus der ermittelten Topologie wird ein *Directed Graph* (gerichteter Graph) erzeugt, aus dem – in diesem Fall für die *Area 2* – die Topologiedatenbasis für den *Area Boundary Router RT4* abgeleitet wird. Aus dem Graphen geht hervor, dass teilweise mehrere Routen zu ein und demselben Ziel existieren, allerdings zu unterschiedlichen Kosten. Routen mit geringeren Kosten werden natürlich bevorzugt. Sind die Kostenwerte jedoch gleich, so werden die Datenpakete zu gleichen Teilen über die gleichwertigen Routen verteilt (*Load Balancing*). Abbildung 4-9 zeigt das Beispiel in anschaulicher Form.

Durch den Lernprozess des Routers *RT4* nach Empfang zahlreicher *Link State Advertisements* lässt sich nicht nur die Topologiedatenbasis seiner *Area 2* bilden, sondern auch eine entsprechende Routing-Tabelle, die auszugsweise wie folgt aussieht:

Zielnetz?	Nächster Hop	Entfernung
N1	RT3	7
N2	RT3	4
N3	–	2
N4	RT6	5
N5	RT5	4
N6	RT3	6

N7

RT3

9

8	RT3
7	RT2
7	N2
7	RT1
10	N1
5	RT9
5	N6
2	RT7
5	RT8
8	N7
2	N5

RT3	1
RT2	2
N2	4
RT1	4
N1	7
RT9	4
N6	6
RT7	6
RT8	6
N7	9
N5	8

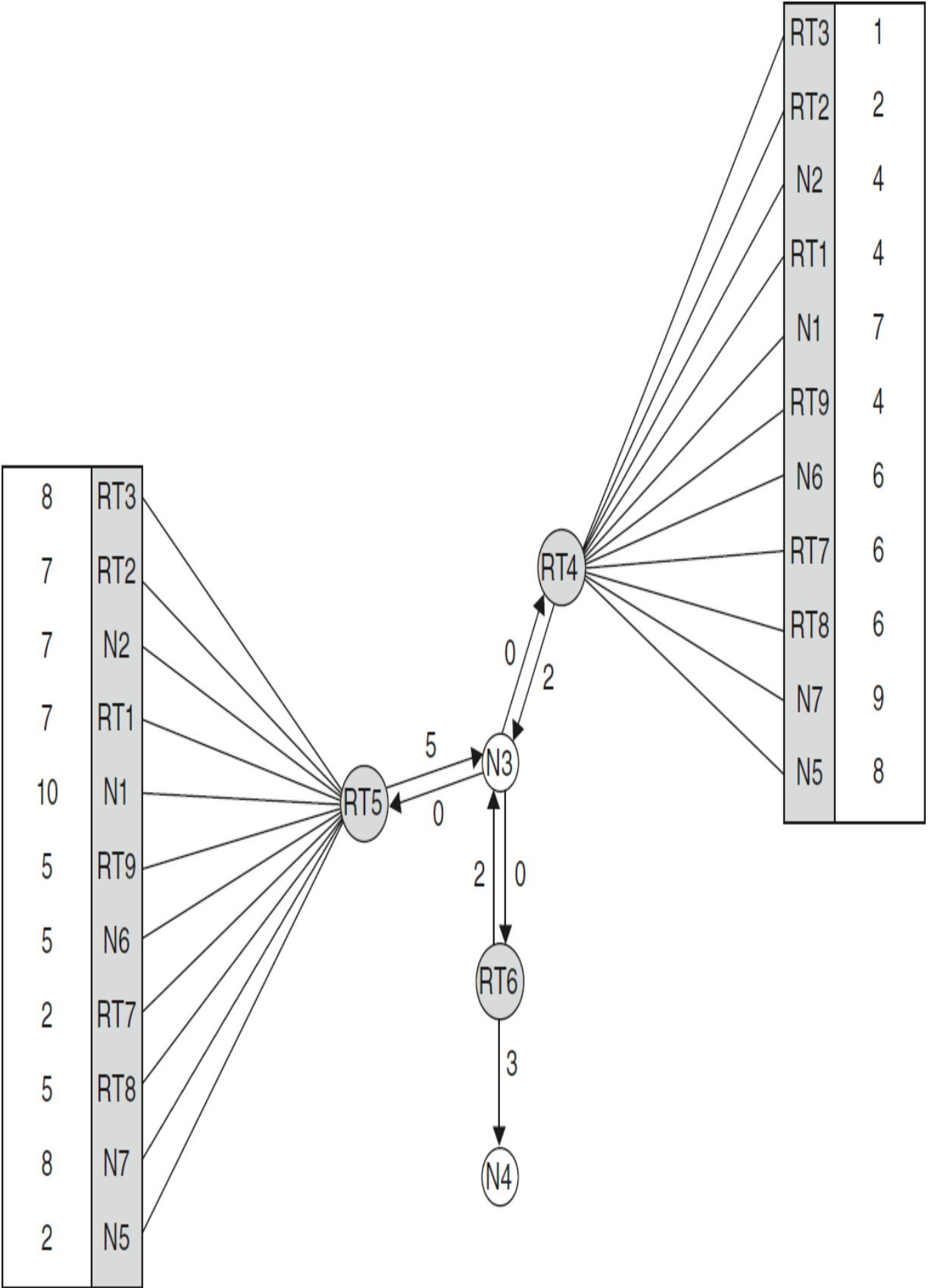
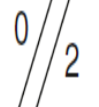
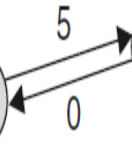


Abb. 4–9 *OSPF-Topologiedatenbasis Router RT4, Area 2*

HINWEIS

Da es sich beim Router RT4 um einen *Area-Border-Router* handelt, verwaltet er mindestens zwei Topologien. In Abhängigkeit vom Feld *Type Of Service* lassen sich separate Routing-Tabellen verwalten. Das aus dem IP-Header verfügbare TOS-Feld ermöglicht ein Routing gemäß den dort hinterlegten Prioritäten.

Parametrisierung

Eine äußerst wichtige, aber auch zeitaufwendige Tätigkeit stellt die Konfiguration eines OSPF-Routers dar. Von den im RFC 2328 zur Verfügung gestellten Parametern sind einige konstant vorgegeben, andere hingegen müssen zunächst einmal in ihrer Bedeutung und Wirkungsweise analysiert werden, bevor man diese modifiziert. In vielen Fällen geben die Router-Hersteller auch Empfehlungen (Default-Werte), die man für die meisten Konfigurationen übernehmen kann.

Die wichtigsten, nachfolgend beschriebenen Parameter sind im Wesentlichen dem RFC 2328 von April 1998 entnommen:

- *LSRefreshTime*

Maximalzeitintervall, nach dessen Ablauf ein neues *Link State Advertisement* generiert wird, sofern kein anderer Grund vorliegt (z.B. Topologieänderung), ein LSA zu versenden. Dieser Timer besitzt den Wert 30 Minuten. Länger darf ein LSA nicht auf sich warten lassen.

- *MinLSInterval*

Mindestzeitintervall, nach dessen Ablauf ein neues LSA generiert werden kann. Der Wert dieses Timers lautet auf 5 Sekunden. Kürzere Intervalle sind nicht erlaubt.

- *MinLSArrival*

Mindestdauer, die zwischen Empfang neuer LSA-Instanzen verstreichen muss. LSAs in kürzeren Intervallen werden ignoriert. Der Default-Wert lautet 1 Sekunde.

- *MaxAge*

Maximalalter eines LSA. Hat ein LSA den Wert in *MaxAge* (= 60 Minuten) erreicht, so wird es für die Berechnung der Routing-Tabelle nicht mehr herangezogen. *MaxAge* muss immer größer als *LSRefreshTime* sein.

- *CheckAge*

Sobald das Alter eines in der Datenbank abgelegten LSA ein Vielfaches des Wertes von *CheckAge* erreicht hat, wird eine Überprüfung der LSA-Prüfsumme vorgenommen. Eine fehlerhafte Prüfsumme weist auf einen schweren Fehler hin. Der Wert steht auf 5 Minuten.

- *MaxAgeDiff*

Maximalzeit, in der ein LSA das AS durchqueren darf. Die meiste Zeit warten LSAs in den Routern bzw. ihren Ausgabepuffern (in dieser Zeit »altern« sie jedoch nicht). Dieser Parameter besitzt den Wert 15 Minuten.

- *LSInfinity*

Dieser Link-State-Metrikwert gibt an, ob ein Ziel nicht erreichbar ist. Das Metrikfeld wird dann mit binären Einsen gefüllt (16 Bits bei normalen LSAs, 24 Bits bei Summary LSAs und AS External LSAs).

- *DefaultDestination*

Dieser Parameter enthält die Destination-ID der Default Route. Sie wird verwendet, wenn kein anderer passender Routing-Eintrag gefunden werden kann. Für diese Route werden jedoch nur AS External LSAs und Summary LSAs des Typs 3 benutzt. Der Wert der IP-Adresse lautet 0.0.0.0.

- *InitialSequenceNumber*

LS-Sequence-Number-Wert, der für die zuerst generierte Instanz jedes LSA verwendet wird; Wert = 0x80000001 (signed 32-bit integer)

- *MaxSequenceNumber*

Maximalwert, den eine LS Sequence Number annehmen kann; Wert = 0x7fffffff (signed 32-bit integer)

- *Router-ID*

Eindeutige Router-Kennung. Meist wird hier die niedrigste oder auch höchste IP-Adresse des Routers verwendet. Wird diese geändert, so muss der Router neu gebootet werden, damit die Änderung auch aktiv wird.

- *Area-ID*

In diesem Parameter wird ein 32-Bit-Wert zur Identifikation der Area abgelegt. Die Area-ID mit dem Wert 0 ist für den Backbone reserviert. Repräsentiert die Area ein Subnetz, so lässt sich die Subnetz-ID verwenden.

- *List of address ranges*

Eine OSPF-Area ist als Liste von IP-Adress- und -Maskenpaaren definiert. Jedes Paar beschreibt einen IP-Adressbereich. Netzwerke und Hosts werden gemäß ihrer IP-Adresse und dem entsprechenden Bereich einer Area zugeordnet. Router gehören meist mehreren Areas an, je nach zugeordneten Netzwerken.

- *External routing capability*

Dieser Parameter gibt an, ob *AS External Advertisements* in oder durch die eigene Area transportiert werden dürfen. Ist dies nicht erlaubt, so handelt es sich um eine »Stub Area«. Innerhalb eines solchen Bereichs erfolgt das Routing zu externen Zielen über Default Routes. Die Backbone Area darf niemals Stub Area sein. Virtuelle Links dürfen nicht über Stub Areas definiert werden.

- *StubDefaultCost*

Bei einer *Stub Area* gibt ein Area-Border-Router dieser Area über einen Parameter die Kosten der Default Route bekannt.

- *IP interface address*

Hier ist die netzwerkweit eindeutige IP-Adresse der Router-Schnittstelle anzugeben. Serielle Leitungen können als *unnumbered* gekennzeichnet werden und auf die IP-Adresse verzichten.

- *IP interface mask*

Angabe der Subnetzmaske des angebundenen Netzwerks

- *Interface output cost(s)*

Es werden die Kosten konfiguriert, die für den Datentransport über diese Router-Schnittstelle berechnet werden sollen. Diese Werte sind Bestandteil der Router-LSAs. Für jeden TOS dürfen verschiedene Kostenwerte angegeben werden.

- *RxmtInterval*

Gibt die Zeit an, die zwischen zwei LSA-Retransmissions liegen muss. Der Wert sollte deutlich über dem *round trip delay* zwischen zwei beliebigen Routern im Netzwerk liegen. Auf seriellen Leitungen muss dieser Wert entsprechend höher konfiguriert werden. In LANs wird ein Wert von 5 Sekunden empfohlen.

- *InfTransDelay*

Gibt an, wie lange es dauert (in Sekunden), ein LS-Update-Paket über diese Schnittstelle zu versenden. LSAs, die in dem Update-Paket enthalten sind, müssen den Wert für ihr »Alter« um diesen Wert erhöhen, bevor sie übertragen werden. Dieser Wert sollte die Verzögerungszeiten der Schnittstelle berücksichtigen. Der Wert muss größer als 0 sein. Der Wert 1 wird empfohlen.

- *Router priority*

Dieser Wert wird für die Bestimmung des *Designated Router* herangezogen. Der Router mit dem höchsten Wert »gewinnt«. Bei gleichem Wert entscheidet die höhere Router-ID. Ein Router mit der Router Priority von 0 kann niemals *Designated Router* werden.

- *HELLOInterval*

Zeitintervall, in dem *HELLO*-Pakete vom Router über die Schnittstelle versendet werden. Innerhalb eines Netzwerks muss dieser Parameter auf allen Routern identisch sein. Je kleiner das Intervall, desto schneller werden topologische Änderungen ermittelt, umso höher ist jedoch auch die Netzbelastung. Empfohlene Werte für ein X.25-Netzwerk sind 30 Sekunden, für ein LAN 10 Sekunden.

- *RouterDeadInterval*

Gibt die Zeit (in Sekunden) an, die seit dem letzten *Hello* dieses Routers verstrichen ist, bevor die Nachbar-Router ihn

als *down* bzw. inaktiv deklarieren. Das Vierfache des *HELLOInterval* wird empfohlen. Es ist darauf zu achten, dass dieser Wert für alle im Netzwerk involvierten Router identisch ist.

- *Autype*

Jeder Area kann ein separater Authentifizierungstyp zugeordnet werden. Über diesen Mechanismus müssen sich Router identifizieren, wenn sie am OSPF-Routing teilnehmen wollen. Drei mögliche Werte sind implementierbar: 0 = keine Authentifizierung, 1 = 64-Bit-Kennwort muss angegeben werden, 2 = Paket-Verschlüsselung (Einzelheiten siehe RFC 2328, Anhang D).

- *Authentication key*

Jeder am OSPF teilnehmende Router weist seine Berechtigung durch dieses Kennwort nach.

- *List of all other attached routers*

Beschreibt eine Liste aller übrigen im Non-Broadcast-Netzwerk integrierten Router. Sie werden mit ihrer IP-Adresse angegeben. Ferner ist ersichtlich, ob der entsprechende Router *Designated Router* werden kann.

- *PollInterval*

Wenn ein Nachbar-Router inaktiv wird (*RouteDeadInterval* abgelaufen), ist es ggf. erforderlich, *HELLO*-Messages an ihn zu verschicken. Diese *Hellos* werden jedoch in einem

verringerten Poll-Intervall gesendet, das erheblich größer sein soll als das *HELLOInterval*. Für X.25-Netzwerke wird ein Wert von 2 Minuten empfohlen.

- *Host IP address*

IP-Adresse des direkt erreichbaren Hosts

- *Cost of link to host*

Kostenwert für den Versand eines Datenpakets vom Router zu diesem Host (auch hier darf für jeden TOS ein entsprechender Wert vergeben werden)

- *Area-ID*

Angabe der OSPF-Area, zu der dieser Host gehört

Datagramme

Das OSPF-Datagramm setzt unmittelbar auf das Internet Protocol auf, d.h., dem *IP-Header* folgt unmittelbar das OSPF-Datenpaket. Da im OSPF eine Fragmentierung nicht vorgesehen ist, übernimmt entweder IP diese Funktionalität oder aber es werden von vornherein kleine OSPF-Datenpakete generiert, die ein Fragmentieren überflüssig machen. Die Adressierung der OSPF-Pakete erfolgt über die bereits mehrfach erwähnten Multicast-Adressen aus der IP-Adressklasse D (reserviert). Für das Multicasting werden zwei IP-Adressen verwendet. Die Adresse 224.0.0.5 richtet sich an alle Router. Die über diese Adresse verschickten Multicast-Datenpakete müssen von allen

Routern bearbeitet werden (z.B. *HELLO-Messages*). Alle *Designated Router* mit ihren Backups werden über die Adresse 224.0.0.6 angesprochen. Der *OSPF-Header* setzt sich aus den in [Abbildung 4-10](#) dargestellten Feldern zusammen.

Version	Type	Packet Length
	Router ID	
	Area ID	
Checksum	AuType	
	Authentication	
	Authentication	

Abb. 4-10 *OSPF-Header*

- *Version* (8)
Enthält die OSPF-Versionsnummer.
- *Type* (8)
Hier wird der Typ des OSPF-Datenpakets angegeben, wobei folgende Typen definierbar sind: *HELLO-Paket* (type 1), *Database Description* (type 2), *Link State Request* (type 3), *Link State Update* (type 4), *Link State Acknowledgement* (type 5).
- *HELLO-Paket* (type 1)
Das periodisch generierte *HELLO-Paket* sorgt für den Aufbau und die »Pflege der Nachbarschaft« mit anderen

Routern. Es werden bestimmte Parameter abgeglichen, die im Netzwerk in allen Routern identisch sein müssen. Unterschiede können dazu führen, dass bei der »Nachbarschaftspflege« Probleme auftreten oder ein Aufbau der Nachbarschaften (*Adjacencies*) erst gar nicht zustande kommt.

- *DATABASE DESCRIPTION (type 2)*

Die DD-Pakete werden zum Aufbau einer Topologiedatenbank benötigt. Im Master-Slave-Verhältnis werden im Polling-Verfahren entsprechende *Link State Advertisements* (LSA) ausgetauscht. Alle fünf LSA-Typen besitzen einen Header, der aus den Feldern *LSAge*, *Options*, *LSType*, *LinkStateID*, *Advertising Router*, *LSsequence number*, *LSchecksum* und der Länge des gesamten LSA besteht. Anschließend werden die fünf unterschiedlichen »Bodies« der LSAs angehängt.

- *LINK STATE REQUEST (type 3)*

Sollte nach mehrfachem Austausch von Database-Description-Paketen festgestellt werden, dass ein anderer Router aktuellere Informationen besitzt, werden zur Aktualisierung gezielte Link State Requests verschickt. Diese bestehen aus dem *LSType*, der Link State ID und dem Advertising Router.

- *LINK STATE UPDATE (type 4)*

Mit einem *Link State Update* werden die *Link State Advertisements* (LSAs) übertragen. Innerhalb eines LSU können mehrere LSAs übertragen werden. Die genaue Anzahl geht aus dem Feld *number of advertisements* hervor.

- *LINK STATE ACKNOWLEDGEMENT (type 5)*

Zur Sicherung des Datenflusses wird für die *Link State Updates* ein Bestätigungsverfahren eingeführt, das jedes einzelne oder auch mehrere LSAs gemeinsam über Multicasts bestätigt.

- *Packet Length (16)*

Länge des OSPF-Pakets in Oktetten (Bytes)

- *Router ID (32)*

Gibt die eindeutige Identifikation (IP-Adresse) des sendenden Routers an.

- *Area ID (32)*

Gibt die *Area* an (meist Subnetzadresse), zu der das Paket gehört. Alle OSPF-Pakete sind einer bestimmten Area zugeordnet, die zumeist höchstens einen Hop vornehmen. Pakete, die über einen virtuellen Link übertragen werden, erhalten die Backbone-Area-ID 0.

- *Checksum (16)*

Prüfsumme, die aus dem gesamten Paket mit Ausnahme des Authentifizierungsfelds berechnet wird

- *AuType* (16)

Hier wird der Authentifizierungstyp abgebildet, der aktuell verwendet werden soll. Zwei Werte sind zurzeit definiert: Ein *AuType* von 0 gibt an, dass keine Authentifizierung erfolgt, ein *AuType* von 1 weist auf die Benutzung eines maximal acht Oktette umfassenden Kennworts hin, ein *AuType* von 2 ermöglicht Verschlüsselung.

- *Authentication* (64)

Hier wird, entsprechend dem *AuType*, das Kennwort hinterlegt.

OSPF-Weiterentwicklung

Auch wenn OSPF in der Version 2 heute noch überwiegend in Netzwerken anzutreffen ist, so lohnt doch ein Blick auf die Weiterentwicklung des populären Routing-Protokolls der letzten Jahre. Diese sind dem öffentlich verfügbaren RFC-Index zu entnehmen.

Hieraus ist ersichtlich, dass es mittlerweile eine OSPFv3 gibt, die sich grundsätzlich von OSPFv2 dahin gehend unterscheidet, dass sie der IPv6-Technologie Rechnung trägt.

HELLO

Basierend auf einer Implementierung von PDP-11-Software wird *HELLO* innerhalb des *Distributed Computer Network* (DCN) als Routing-Protokoll verwendet. Es ist dem RIP-Protokoll verwandt, benutzt allerdings keine *Hop-Counts* zur Routenoptimierung, sondern arbeitet nach dem Delay-Konzept, das primär die Verzögerungszeiten in einem Netzwerk zugrunde legt und daher als wesentlich realistischer im Vergleich zu RIP eingestuft werden kann. Es entspricht auch eher dem Empfinden der Netzanwender, Antwortzeiten zu berücksichtigen, als einfache *Hop-Counts*, die unterschiedlichen LAN- bzw. Leitungsgeschwindigkeiten keinerlei Beachtung schenken.

Für eine zeitgesteuerte Routing-Konzeption ist allerdings ein exakter Synchronisationsprozess auf allen Routern erforderlich. Dies bedeutet, dass mit einem erhöhten Overhead im Netzwerk und damit mit einer deutlichen Netzwerkbelastung gerechnet werden muss. Jedes Datenpaket wird beim Router-Durchlauf mit einem *Time Stamp* versehen, der vom nächsten Router gelesen und interpretiert wird. Infolge der ermittelten Zeitabweichung zu seiner eigenen Uhrzeit kann der Router das für die optimale Route relevante *Delay* berechnen und entsprechend für seine Auswahl

zugrunde legen. Aus der Vielzahl periodisch versendeter bzw. empfangener *Hellos* wird die aktuelle Zeit im Netzwerk kontinuierlich untereinander abgestimmt und dadurch aktualisiert.

Zur Vermeidung eines *Route Hopping* wird eine Schwellenwertzeit beim Delay-Update eingesetzt. Ein unmittelbares Update hätte zur Folge, dass bei einer Routenänderung im Netzwerk *alle* Router die neue, günstigere Route verwenden und dadurch eine Überlastung dieser Route hervorrufen würden. Die Überlastung führte allerdings dann wieder zu einem Verlassen der Route (sie wird ungünstiger), und anschließend würde diese erneut bevorzugt. Dieser *Hopping*-Vorgang würde sich kontinuierlich wiederholen.

Interior Gateway Routing Protocol (IGRP)

Neben allgemeinen Routing-Protokollen, die nicht aus den Labors bestimmter Hersteller stammen, hat es der Router-Hersteller Cisco geschafft, sein eigenes proprietäres Routing-Protokoll *Interior Gateway Routing Protocol* (IGRP) in seinen Routern auf dem Markt zu platzieren. Als die Komplexität der Netzwerke Mitte bis Ende der 80er Jahre des vorigen Jahrhunderts deutlich zunahm und es lediglich das für diese Zwecke etwas überforderte *Routing Information Protocol* (RIP)

gab, empfahl sich IGRP als leistungsfähige Alternative. Während RIP den Hop-Count als einzig relevante Metrik verwendet, benutzt IGRP eine Kombination mehrerer Metriken. So sind beispielsweise das *Internetwork Delay*, also der Verzögerungsfaktor zwischen einzelnen Netzwerken, sowie Überlegungen zur garantierten Bandbreite vom Netzwerksystemverwalter zu bewertende Metriken und dienen ihm daher als Instrumentarien zur gezielten Einflussnahme auf die Wahl der Routen. Der Aufbau alternativer Routen im Falle von Störungen gehört ebenso zum Funktionsumfang wie die Lastverteilung; eine Route, die z.B. nur halb so teuer ist wie andere, wird auch doppelt so oft benutzt.

HINWEIS

Wie im OSPF ist auch im IGRP das Verhalten der einzelnen Router im Netzwerk über verschiedene *Timer* steuerbar. Es gibt *update timer*, *invalid timer*, *flush timer* oder auch *hold-time periods*.

Die Weiterentwicklung von IGRP wurde von Cisco in dem unmittelbaren Nachfolger EIGRP (Enhanced IGRP) fortgesetzt; siehe nachfolgender Abschnitt.

Enhanced IGRP

Anfang der 90er Jahre des vorigen Jahrhunderts wurde in den Routern der Firma Cisco die Weiterentwicklung des IGRP implementiert: Enhanced IGRP. Es stellt eine gelungene Mischung aus *Link-State-Protokollen* (wie beispielsweise das OSPF) und *Distance-Vector-Protokollen* (wie RIP oder auch IGRP) dar. Folgende Eigenschaften werden unterstützt:

- Durch einen neuen Algorithmus beim Update von Routing-Informationen (*Diffusing Update Algorithm* = DUAL) werden die Routing-Tabellen der Nachbar-Router ebenfalls gespeichert und stehen somit für eine rasche Alternativroutenwahl unmittelbar zur Verfügung. Sollte dennoch eine Route nicht zustande kommen, so wird diese vom Nachbarn erfragt.
- Verwendung variabler Subnetzmasken (variabel in ihrer Länge), wodurch eine Bündelung von Subnetzrouten am logischen (Netz-ID des Subnetzes) Netzwerkübergang erfolgen kann
- Durchführung begrenzter Teil-Updates, bei denen Updates nicht periodisch wiederkehrend vorgenommen werden, sondern nur dann, wenn sich der Status der Metrik ändert. Nur diejenigen Router werden mit den relevanten Updates versorgt, die sie auch benötigen. Eine deutlich reduzierte

Netzbelastung durch *Enhanced IGRP* gegenüber IGRP ist die Folge.

- Es lassen sich AppleTalk, IP und beispielsweise IPX gemeinsam verwenden. Dabei bezieht AppleTalk seine Routing-Informationen aus dem *Routing Table Maintenance Protocol* (RTMP), IP erhält die Informationen von OSPF, RIP, IS-IS, EGP oder BGP, während schließlich IPX vom *Novell RIP* und dem *Service Advertisement Protocol* (SAP) bedient wird.
- Einführung des *Reliable Transport Protocol* (RTP), bei dem nur speziell gekennzeichnete (Kennzeichen-)Pakete vom Empfänger bestätigt werden, alle übrigen bleiben unbestätigt.

Intermediate System – Intermediate System (IS-IS)

Im Gegensatz zu anderen Routing-Protokollen entstammt das IS-IS (*Intermediate System – Intermediate System*) als OSI-Protokoll den Bestrebungen der *International Standardization Organization* (ISO). Es gehört wie OSPF zur Kategorie der *Link State Protocols*, basiert auf der DECnet-Architektur *Phase V* und wird auf dem ISO-Netzwerk CLNP (*Connectionless Network Protocol*) eingesetzt. Folgende Protokolle sind dabei definiert:

- ISO 9542 – ES-IS
- ISO 10589 – IS-IS Intradomain Routing Exchange Protocol

- ISO 10747 – IS-IS Interdomain Routing Protocol

HINWEIS

Ein ES (*End System*) ist ein Netzknoten, der keine Routing-Aufgaben übernimmt, und mit IS (*Intermediate System*) wird ein Router bezeichnet.

Die erweiterte Terminologie bilden die *Area* (Zusammenschluss mehrerer Hosts in verschiedenen Netzwerken) und die *Domain* (Verknüpfung von Areas). *Level 1 Router* stellen die Router-Verbindungen innerhalb einer Area her, wohingegen für die Area-zu-Area-Verbindung innerhalb einer Domain *Level 2 Router* zuständig sind.

End System to Intermediate System Protocol (ES-IS)

Bei dem Routing-Protokoll ES-IS handelt es sich mehr um ein Protokoll für das *Routing Discovery* als um ein vollwertiges Routing Protocol. Auf beiden beteiligten Systemen stattfindende Lernprozesse führen zur Bildung von beidseitigen Information-Pools. Es werden drei verschiedene Subnetztypen unterschieden:

- *Point-to-Point-Subnetze*

Punkt-zu-Punkt-Verbindungen zwischen zwei Netzwerken

- Beispiel: serielle oder ISDN-Verbindungen im WAN
- *Broadcast-Subnetze*
Nachrichtenverteilung an alle Knoten im Netzwerk
 - Beispiel: Ethernet, IEEE 802.3, IEEE 802.5
- *General-Topology-Subnetz*
Dieses Subnetz unterstützt eine willkürliche Anzahl von Endsystemen und Routern, ermöglicht jedoch keine Einrichtung zur komfortablen und verbindungslosen Übertragung an mehrere Zielknoten. Man unterscheidet *multipoint links* mit einem Primary-Secondary- bzw. Master-Slave-Kommunikationsverhalten, *permanent point-to-point links* in Form von Standleitungen und *dynamic data links*, die verbindungsorientiert arbeiten, wie beispielsweise X.25, X.21 oder ISDN.

Sowohl ES als auch IS senden sich untereinander *HELLO-Messages* (ESHs und ISHs) zu, um die Konfiguration betreffende Informationen zu übermitteln. Auf Broadcast-Netzwerken geschieht dies unter Verwendung einer Multicast-Adresse. In General-Topology-Netzwerken wird eine Übertragung solcher Konfigurationsdaten vermieden, denn bei (langsamen) WAN-Verbindungen schlägt eine erhöhte Netzlast mit hohen Leitungskosten bzw. einer geringen verbleibenden Bandbreite deutlich zu Buche.

Border Gateway Protocol (BGP)

Beim *Border Gateway Protocol* handelt es sich um ein Protokoll, das nicht zur Kategorie der *Internal Gateway Protocols* (IGP) gehört, sondern auf dem *External Gateway Protocol* (EGP) basiert (siehe RFC 827 vom Oktober 1982).

BGP stellt eine weit flexiblere und leistungsfähigere Alternative zum mittlerweile veralteten EGP dar. Es ermöglicht die Verwendung alternativer Routen zwischen zwei Domänen bzw. Autonomen Systemen (AS) und bietet Mechanismen zur Ermittlung und Vermeidung von Endlosschleifen (*Routing Loops*). Ähnlich wie bei OSPF werden keine vollständigen Routing-Tabellen ausgetauscht, sondern die Aktualität der Tabellen wird durch Updates und *keep alive messages* gewahrt. Während EGP bei überlasteten Netzwerken (*Congestions*) eine sehr schlechte Performance aufweist, arbeitet BGP durch TCP-basierende Retransmissions wesentlich toleranter.

Ein aktuelles Beispiel der auch heute noch betriebenen Weiterentwicklung von BGP (mittlerweile in Version 4 – gemäß RFC 4271 vom Januar 2006) kann dem RFC 6774 vom November 2012 entnommen werden. Hier wird ein relativ einfaches Verfahren zur BGP-Optimierung beschrieben, das im Wesentlichen durch eine Anpassung der BGP-Konfiguration auf

Routern erreicht werden kann. Das BGP-Protokoll selbst wird in seiner Spezifikation dabei nicht verändert.

Betrieb und Wartung

Für den Praxiseinsatz lassen sich beim Betrieb und der Wartung von Routern folgende Schwerpunkte definieren:

- Diagnose von Hardwareproblemen des Router-Prozessors, seiner Netzwerkschnittstellen, seiner angebundenen Netzwerke, Modems oder Kommunikationsleitungen
- Installation neuer Hardware
- Installation neuer Software
- Neustart oder Reboot eines Routers nach erfolgtem Ausfall
- Konfiguration oder Rekonfiguration des Routers
- Ermittlung von IP-Problemen wie Überlast, Routing-Schleifen, fehlerhafte IP-Adressen, Broadcast-Stürme oder Fehlverhalten von Hosts
- Modifikation der Netzwerktopologie, entweder temporär (Aufbau alternativer Routen bei vorübergehenden Leitungsproblemen im Netzwerk) oder permanent
- Erstellung von Netzwerkstatistiken, um eine geeignete Netzwerkplanung zu realisieren
- Koordination vorgenannter Aktivitäten mit geeigneten Herstellern und Spezialisten aus dem Bereich der

Telekommunikation

In vielen Unternehmen oder Organisationen erfolgt das Router-Management an zentraler Stelle und ausgelegt von einem gut ausgebildeten Team von Netzwerkspezialisten, die den oder die Router in der Regel über Netzwerkverbindungen installieren, konfigurieren und steuern. Dabei beschränkt sich der Zugang oft nicht auf eine klassische Netzwerkverbindung, sondern es werden separate Wählleitungen installiert, die auch bei Netzwerkstörungen den Router-Zugang ermöglichen. Dabei müssen Hilfsmittel für das Netzwerkmanagement und für die Beobachtung des Netzwerks und seiner Komponenten (Monitoring) verfügbar sein, wobei »überflüssiges« Monitoring immer zu einer zusätzlichen Belastung des Netzwerks führt; es gilt hier also, ein ausgewogenes Verhältnis zu finden.

Router-Initialisierung

Soll ein Router in ein Netzwerk integriert werden, so ist dieser in der Regel mit der Basissoftware bzw. seinem Betriebssystem vorinstalliert. In vielen Fällen ist dann auch bereits eine mit Default-Werten versehene Konfiguration vorhanden, die allerdings immer den eigenen Bedürfnissen angepasst werden muss. Fehlt beispielsweise eine Definition von IP-Adressen oder Subnetzmasken für einzelne Netzwerkschnittstellen, so sind bei

einer Inbetriebnahme des Routers keinerlei Probleme zu erwarten. Geht der Router jedoch mit voreingestellten IP-Adressen ins Netz, so kann dies zu nicht vorhersehbaren Problemen führen. Eine individuelle Konfiguration sollte daher immer zu den ersten Aktivitäten vor Inbetriebnahme eines Routers gehören.

Die Funktionsfähigkeit eines Routers lässt sich grundsätzlich auf zwei verschiedenen Wegen herstellen: Entweder erfolgt die Installation bzw. Konfiguration manuell, wobei diese Informationen permanent gespeichert werden, sodass der Router nach Störungen jederzeit wieder rekonfigurierbar ist, oder seine vollständige Systemsoftware wird über das Netzwerk durch einen dedizierten Boot-Rechner unter Verwendung des BootP- und TFTP-Protokolls zur Verfügung gestellt.

Wie in [Kapitel 3](#) (siehe dort) ausführlich dargestellt, ist BootP ein Protokoll, das zur Ermittlung von Rechnern dient, von denen ein Endsystem sein Betriebssystem und Konfigurationsparameter bezieht. Zu diesem Zweck wird vom Router ein *BootP Request* generiert, der die Hardwareadresse der eigenen Netzwerkschnittstelle enthält. Derjenige BootP-Server, der diese empfangene Hardwareadresse kennt, übermittelt dem Router die ihm bekannt gegebene IP-Adresse

des Routers, seine eigene IP-Adresse und den Namen des *Bootfile Image*. Nach Empfang dieser *BootP Response* (und ihrer lokalen permanenten Speicherung) besitzt der Router alle erforderlichen Informationen, um den Boot-Vorgang als vollwertiger IP-Knoten mit seinem Kommunikationspartner durchführen zu können, wozu zusätzlich auch ein File-Transfer-Protokoll benötigt wird; in den meisten Fällen ist dies TFTP (*Trivial File Transfer Protocol*).

HINWEIS

Auch bei einer automatisierten Installation und Konfiguration eines Routers ist es erforderlich, dass ein Router nicht nur direkt steuerbar ist – z.B. über eine angeschlossene Konsole oder über eine lokal bedienbare proprietäre Tastatur –, sondern auch über eine Netzwerkverbindung kontrolliert werden kann. Die erforderliche Kommunikation sollte dann über Standardprotokolle bzw. -anwendungen wie TELNET, FTP oder SNMP möglich sein.

Out-Of-Band Access

Bei anhaltenden Fehlersituationen in einem Netzwerk ist es oft nicht mehr möglich, einen Router über die Netzwerkverbindung (z.B. per *TELNET*) zu erreichen. Eine

Fehleranalyse (Protokolldateien) oder ein Absetzen wichtiger Kommandos kann dann nicht mehr vorgenommen werden, sodass ein *direkter* Zugang zum Router notwendig wird. Wenn sich der Router jedoch außerhalb des direkten Einzugsbereichs befindet und möglicherweise in einer anderen Location (viele Kilometer entfernt) untergebracht ist, würde sehr viel Zeit verstreichen, bis man vor Ort ein entsprechendes Operating durchführen könnte.

Für solche Notfälle ist es zu empfehlen, einen *Remote Access* durchzuführen, der vom eigentlichen Netzwerk unabhängig implementiert ist. Dieser *Out-Of-Band Access* (OOB) lässt sich beispielsweise durch eine separate Wählleitung (analog oder digital) über einen ISDN-Controller realisieren, wobei natürlich für diese Art der Fernwartung in jedem Router eine zusätzliche Schnittstelle eingebaut und konfiguriert werden muss (siehe [Abb. 4-11](#)).

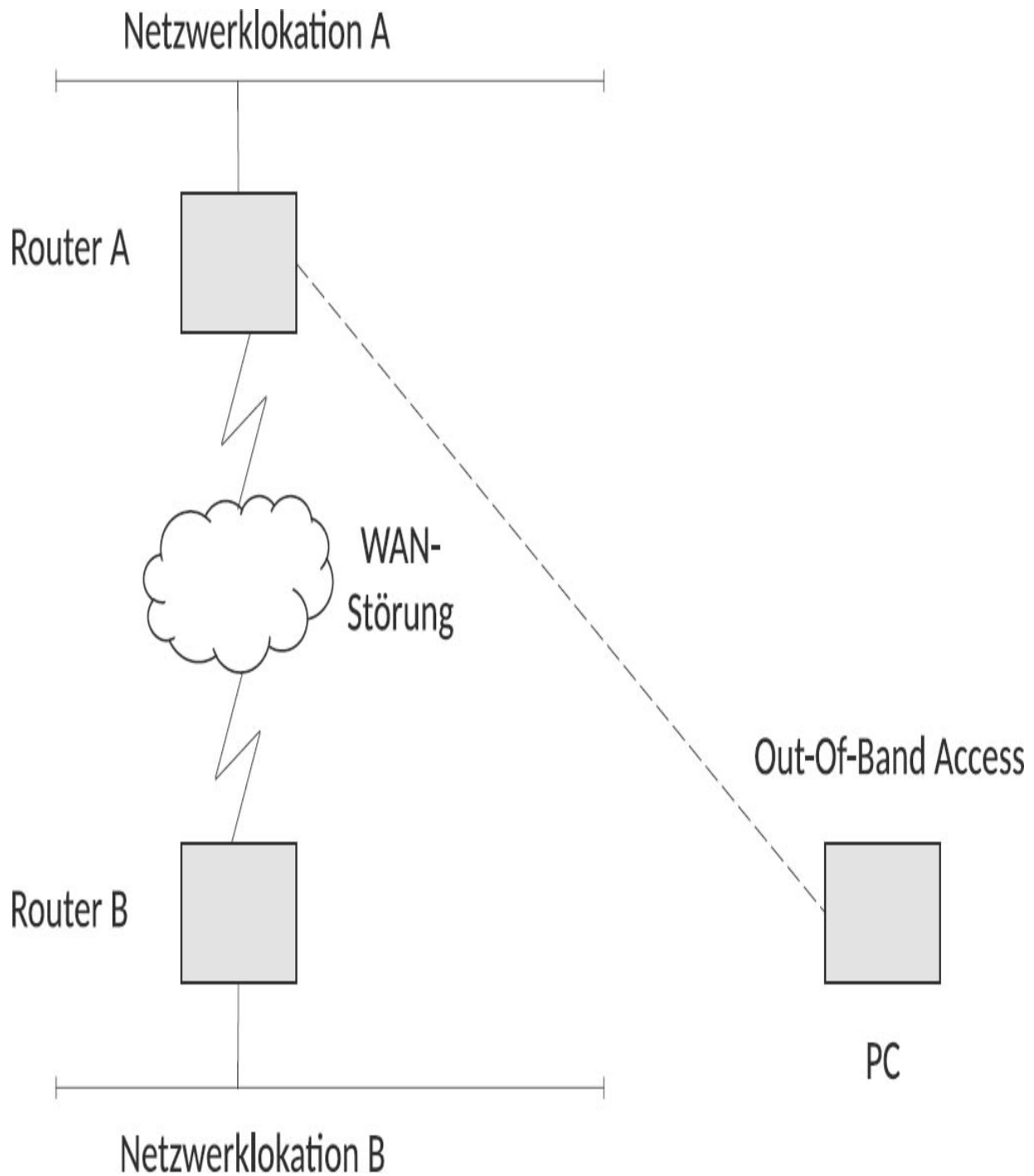


Abb. 4-11 *Out-Of-Band Access*

Heute arbeitet man auch zunehmend mit mobilfunknetzbasierenden OOB-Komponenten (GSM-Modems), die

in einen Router »eingeschleift« werden können. Man schafft sich somit die Möglichkeit, einen Router im Störfall vom Stromnetz zu trennen, und ist nicht nur auf *Software-Resets* beschränkt.

HINWEIS

Beim Aufbau eines (oben beschriebenen) »Notfallzugangs« ist darauf zu achten, dass die OOB-Verbindung keinerlei Einschränkungen in den wesentlichen Funktionen des Router-Managements unterliegt.

Hardwarediagnose

Für die Überprüfung der rudimentären Betriebs- und Funktionsfähigkeit eines Routers müssen in der Basissoftware Programme zur Verfügung stehen, die Aufschluss über den aktuellen Zustand des Geräts geben können. Die Ausführung von Diagnoseprogrammen oder die Durchführung von Selbsttests (einschließlich aller angeschlossenen Schnittstellen) ist obligatorisch. Ferner sollte es die Möglichkeit geben, einen System-Dump (Hauptspeicherauszug) zu erzeugen.

Router-Steuerung

Für die automatische Wiederherstellung (*Recovery*) in Störsituationen ist ein Automatismus erforderlich, der einen *Reboot* oder einen *Restart* des Routers auslöst. Nicht immer ist es sinnvoll, so lange zu warten, bis das Netzwerkteam den Fehler erkennt, analysiert und anschließend die Fehlersituation beseitigt. Wenn der Router automatisch neu startet, ist es allerdings wünschenswert, dass die Fehlersituation nicht nur lokal in einer Berichtsdatei gesichert, sondern möglichst auch ein Speicherauszug (*Memory Dump*) ausgeführt wird, um Fehler nachträglich analysieren zu können.

Änderungen an der Router-Konfiguration erfordern nicht selten einen Systemstopp bzw. einen Neustart; einige der Konfigurationsparameter werden unmittelbar nach der Einrichtung aktiv, einige erfordern einen Neustart (*Reboot*). Die Planung von Konfigurationsänderungen sollte diesen Sachverhalt berücksichtigen, da ein *Reboot* des Routers stets mit einem Ausfall der betroffenen Netzwerkverbindungen einhergeht. Erfahrungsgemäß empfiehlt sich eine Durchführung dieser Wartungsarbeiten in fest eingeplanten Wartungsintervallen oder, sofern es derartige Wartungsfenster nicht gibt, an Wochenenden oder Feiertagen.

Sicherheitsaspekte

Ein äußerst strittiges Thema ist das *Auditing* und das *Accounting*. Es besteht die aus Sicherheitsgründen verständliche Anforderung, wesentliche Konfigurationsänderungen im Router-Profil zu protokollieren, damit jederzeit nachvollzogen werden kann, wer wann welche Änderungen vorgenommen hat. Verletzungen der Filtering-Vorschriften oder Autorisationsmängel (falsche Passwörter, ungültige SNMP-Communities usw.) gehören zu den *Audit-relevanten* Ereignissen. Darüber hinaus sollte jeder Anwender in dem Maße mit anfallenden fixen und vor allem variablen Kosten belastet werden (z.B. Hardware- und Software-Leasing, Leitungskosten), wie er den Router in Anspruch nimmt; in dem Zusammenhang ist auch von *Packet Accounting* die Rede.

Damit die geforderten Informationen zur Verfügung gestellt werden können, bedarf es einer Fülle von Prozessen, die zusätzlich zum »normalen« Datenverkehr auf dem Router betrieben werden müssen. Die Leistungsfähigkeit des Routers wird dadurch natürlich mehr oder weniger stark beeinträchtigt. In diesem Interessenkonflikt muss versucht werden, ein individuelles, aber ausgewogenes Konzept zu entwickeln, das allen Anforderungen in vernünftigem Umfang entsprechen kann.

Software Defined Networking (SDN)

Heutige Netzwerke verlangen aufgrund des erheblich zugenommenen Datenaufkommens leistungsfähige Routing-Konzepte. Da sich an den eigentlichen Routing-Protokollen in den letzten Jahren nicht viel geändert hat, sind neue Konzepte erforderlich, die eine deutliche Optimierung der Wegewahl innerhalb der Netzwerke ermöglichen. Die Technologie des *Software Defined Networking* (SDN) stellt solch eine Optimierung dar.

Dem SDN liegt formal eine Trennung »der Kontrollebene von der Datenebene« zugrunde. Eine zentrale (Hardware-unabhängige) Instanz übernimmt die vollständige Kontrolle des Netzwerks und all seiner beteiligten Komponenten. Sämtliche erforderliche Kommunikation beispielsweise zur Parameterabstimmung oder Routenoptimierung läuft über diese Instanz. Man reduziert dadurch die Komplexität mehr oder weniger gleichberechtigter Netzwerkkomponenten durch einen »Chef-Kommunikator«, der auch gleichzeitig in einer Hierarchie an oberster Stelle steht.

Mit dieser konzeptionellen Änderung im Netzwerk erhält der Netzwerkadministrator im Unternehmen eine wesentlich bessere Kontrolle über »sein« Netzwerk, da er diese ohne

Hardware-Restriktionen bzw. -Spezifikationen der Hersteller ausüben kann. Open-Source-Software zur Administration von Netzwerken spielt in dieser Umgebung eine besondere Rolle; denn sie ist überall verfügbar und unter Berücksichtigung eines allseits akzeptierten Regelwerks sehr flexibel einsetzbar. Die Zukunft wird nicht mehr von z.T. proprietärer »Netzwerk-Intelligenz« der Hersteller bestimmt, sondern die Kontrolle erhält nun das Unternehmen bzw. der Internet-Service-Provider mit eigenen Entwicklungen und der »Selbst-Programmierung« ihrer Netzwerke.

Abb. 4-12 zeigt eine vereinfachte SDN-Architektur, die allerdings nur das Grundprinzip darstellt, nicht aber alle im SDN beheimatete Varianten, wie sie in den folgenden Abschnitten kurz erläutert werden.

HINWEIS

Die Kontrollebene wird auch *Control Plane*, die Hardware-/Infrastrukturebene wird im internationalen Sprachgebrauch auch *Data Plane* genannt.

OpenFlow bezeichnet eine Standard-Schnittstelle zur Kommunikation zwischen Kontroll- und Datentransportebene im SDN. Sie wird mittlerweile von zahlreichen Herstellern der

»Data Plane«-Komponenten unterstützt. Der Standard wird von der *Open Network Foundation* (ONF) verwaltet.

Software Defined Networking (SDN)

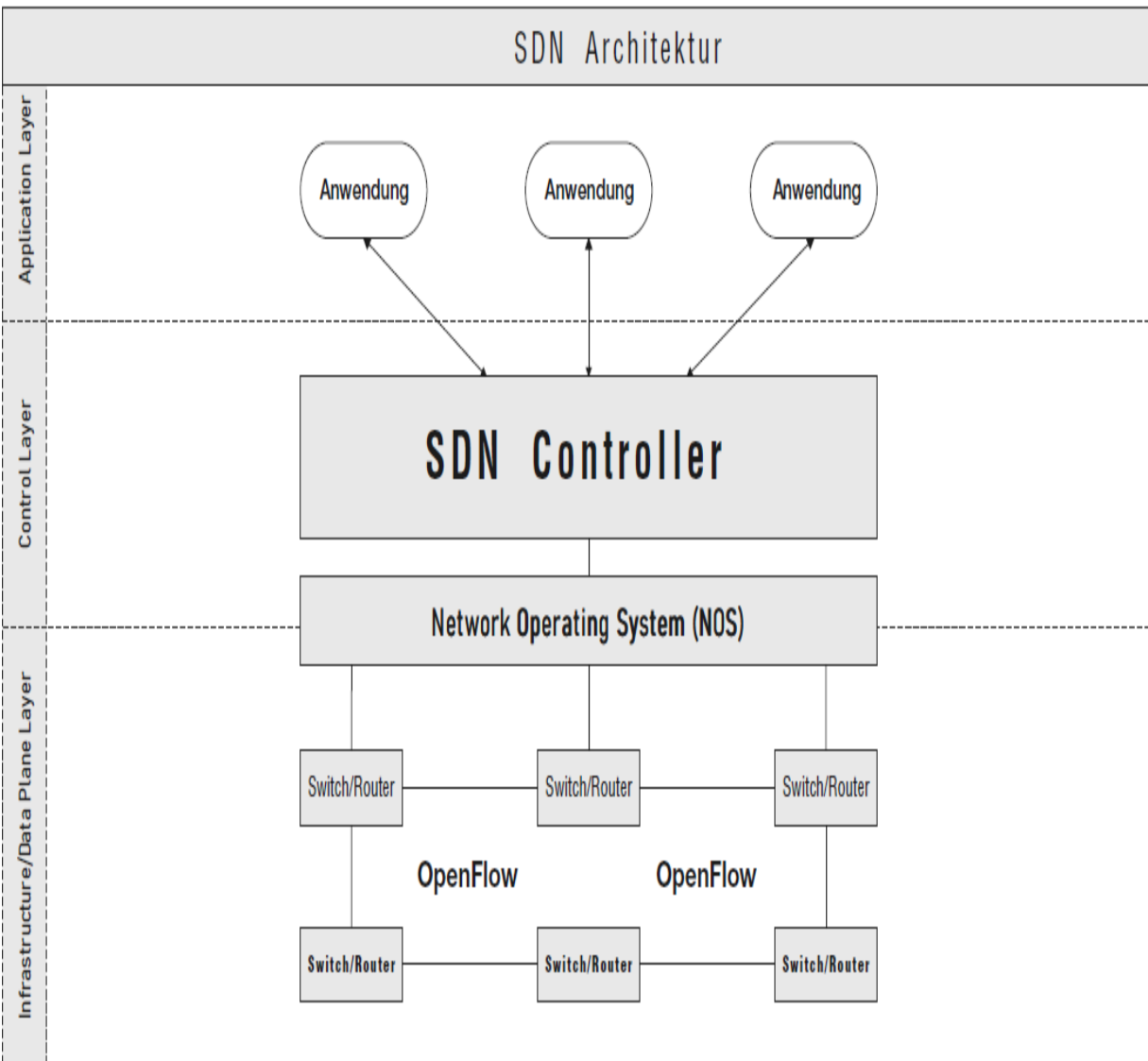


Abb. 4–12 SDN Architektur

SDN ist aber kein Allheilmittel. Es schützt das Netzwerk beispielsweise nicht vor Störungen, die wegen nicht

vorhandener oder unzureichender Transparenz des Netzwerks entstehen. Daher ist es u.a. von entscheidender Bedeutung, dass der »Controller« sämtliche Statusinformationen oder auch Aktivitäten der Netzwerkadministratoren akribisch protokolliert.

Beispiele für SDN-Implementierungen sind:

Netzwerk Virtualisierung

Sie stellt eine weit umfassendere Virtualisierung von Netzwerk-Ressourcen dar, als diese mit den seit Jahren bereits praktizierten virtuellen privaten Netzwerken (VPNs) oder virtuellen LANs (VLANs) möglich sind.

Switching Fabrics

Sie wird primär bei Cloud-Providern und größeren Unternehmensnetzwerken eingesetzt. Dieses Konzept vermeidet den Einsatz von proprietärer Hardware und verwendet stattdessen einfache switching Hardware mit eigener »Intelligenz« über Softwarelösungen. Switching Fabrics kommen oft im Zusammenhang mit der neuen »Spine-Leaf«-Netzwerkarchitektur zum Einsatz, das zunehmend das ältere hierarchische Modell (*Access – Distribution – Core*) ablöst. Die neue Architektur bietet einige Vorteile wie beispielsweise eine

reduzierte Latenz (kürzere Wege in einer vermaschten Zwei-Ebenen-Konzeption) sowie eine bessere Skalierbarkeit (Komponenten können auf beiden Ebenen problemlos ergänzt oder entfernt werden).

WAN Traffic Engineering

Diese Technologie ist prinzipiell nicht neu, sondern wird bereits seit Jahren in MPLS-Netzwerken im WAN angewendet – allerdings ohne die im SDN-Umfeld nun mögliche zentrale Übersicht des Datenverkehrs und das damit verbundene Optimierungspotenzial (z.B. bei der Re-Kalkulation von Routen im Zuge von Statusänderungen einzelner Verbindungen). Die Priorisierung bestimmter Datenklassen stellt eine weitere Funktion zur Effizienzsteigerung des Netzwerks dar.

SD-WAN

Mit SD-WAN wird die klassische Strukturierung eines MPLS-Netzwerks in CE-Routern (*Customer Edge Router*; beim Kunden) und PE-Routern (*Provider Edge Router*; beim Provider/Netzwerkanbieter) durch eine wesentlich flexiblere Konzeption abgelöst. Die zentrale Konfiguration von Routern in Zweigstellen eines Unternehmensnetzwerks erfolgt automatisch, wenn der Router in der Zweigstelle eingetroffen und mit dem Netzwerk verbunden ist. Separater und

individueller Konfigurationsaufwand ist nicht mehr erforderlich. Ferner ermöglichen SD-WAN-Strukturen wesentlich besseres Change Management und reduzieren somit Ausfallzeiten.

Access Networks

Die Implementierung von »Last Mile«-Anbindungen an das Internet werden auch *Access Networks* genannt. So stellt das »Fiber-to-the-Home«-Konzept oder das »radio access network« (4G/5G-Vermittlung) ein solches Access Network dar.

Namensauflösung

Ein entscheidender Aspekt bei der Verwaltung von IP-Adressen stellt die menschliche Bereitschaft (oder Nichtbereitschaft) dar, Zahlen- oder Ziffernkombinationen als Adresskonzept zu akzeptieren. So ist es zweifelsohne einfacher, sich für eine geregelte Kommunikation Namen zu merken, als Ziffern zu verwenden. Das liegt sicherlich nicht zuletzt daran, dass man Namen eine symbolische Bedeutung verleihen kann, im Gegensatz zu Ziffernkombination, mit denen nur sehr schwer symbolische oder gar persönliche Assoziationen verknüpft werden können.

So stellt sich für die Benutzer innerhalb eines Netzwerks bzw. innerhalb eines IP-Systems die Frage, auf welche Art und Weise sie sich die vielen IP-Adressen merken sollen, um beispielsweise bestimmte Endgeräte (Rechner) in einem solchen System direkt ansprechen zu können. Schnell wird dabei klar, dass sich für Anwender Namen wie *EVA* oder *WILLI* wesentlicher besser merken lassen als kryptische IP-Adressen wie *192.112.34.44* oder *156.234.127.167*. Für den täglichen Umgang mit seinem Rechner ist man schon bereit, ihn mit *FINANZ1* oder *VERSAND2* »anzureden«. Dabei spielen solche symbolischen Namen überall dort eine Rolle, wo der Mensch als Kommunikationsinstanz größere Bedeutung hat.

Sich dieser Problematik der schwierigen Zuordnung von IP-Adressen zu Endgeräten bewusst, haben sich die Entwickler des TCP/IP-Protokollstacks schon sehr frühzeitig (Ursprung im ARPANET) ein Verfahren überlegt, mit dem dies umgangen werden kann. Dabei liegt die Lösung im Einsatz eines Systems zur Namensauflösung, mit dem eine Zuordnung von (sprechenden) Namen zu IP-Adressen möglich ist. Oder andersherum ausgedrückt: Den IP-Adressen werden Namen zugewiesen, die sich jeder Anwender besser merken kann und die dann mit einem bestimmten Mechanismus (Namensauflösung) wieder in die zugehörigen IP-Adressen konvertiert werden.

Prinzip der Namensauflösung

Die Problematik bzw. Anforderung der Namensauflösung führte bereits 1987 zu den beiden RFCs 1034 *Domain names – concepts and facilities* und 1035 *Domain names – implementation and specification*. Der RFC 1032 – *Domain Administrator's Guide* beschreibt Verfahren zur Integration lokaler Netze in das weltweite *Domain Name System* (System zur Namensauflösung). Seit dieser Zeit sind gerade zum Thema DNS zahlreiche RFCs veröffentlicht worden, darunter beispielsweise:

- RFC 2136 vom April 1997:Dynamisches DNS durch Einführung eines neuen Update-Message-Typs
- RFC 2219 vom Oktober 1997:Verwendung von Alias-Namen
- RFC 2541 vom März 1999:DNS-Sicherheitsaspekte (Verschlüsselung)

Aber auch in den letzten Jahren wurden zahlreiche RFCs zum Thema DNS veröffentlicht. Hier einige Beispiele zu den Themen:

- Sicherheit – RFCs 9156, 9076, 7218, 6975, 6944, 6781, 6725, 5910, 5702, 4986, 4955 etc.
- Resource Records – RFCs 6742, 5864, 4701
- Mobility Services – RFCs 5679
- IPv6 – 6147, 6106, 5006, 4472
- IANA »Best Practice« – RFCs 9157, 6895
- Multicast – RFCs 7558, 6804, 6742

In einem IP-basierten Netzwerk stehen generell mehrere Verfahren für die Auflösung von Namen zur Verfügung. Neben einem dateibasierten System auf Basis der sogenannten *Hosts-Datei* zählen dazu insbesondere die beiden Verfahren DNS (*Domain Name System*) und WINS (*Windows Internet Naming Service*). Bei DNS und WINS handelt es sich um dynamische

Verfahren zur Namensauflösung, wohingegen der Einsatz einer Hosts-Datei streng statisch ist.

HINWEIS

Voraussetzung für den Einsatz eines Verfahrens zur Namensauflösung ist natürlich, dass das Protokoll TCP/IP zum Einsatz kommt und die Server und Endgeräte (Clients) entsprechend konfiguriert wurden (Netzwerkeigenschaften).

Während DNS als Internetstandard etabliert ist (siehe RFC-Übersicht unter www.rfc-editor.org), stellt WINS eine Sonderlösung von Microsoft dar – quasi eine Umsetzung des Microsoft-proprietären Protokolls »NetBIOS over TCP/IP«.

WINS hat in den vergangenen Jahren jedoch zunehmend an Bedeutung verloren. Heutige Implementierungen stützen sich zumeist auf DNS und greifen nur dann auf WINS zurück, wenn es noch Anwendungen gibt, die NetBIOS-Namen adressieren müssen.

Symbolische Namen

Die Realisierung einer Implementierung symbolischer Namen lässt sich auf zweierlei Wegen vollziehen, wobei eine statische und eine dynamische Methode existieren. Die statische

Methode wird auf Systemen durch die Einrichtung einer Datei (*Hosts*) realisiert.

HINWEIS

In vielen Fällen können neben dem eigentlichen Host-Namen auch noch ein oder mehrere Alias-Namen zu einer IP-Adresse definiert werden. Dies ist dann zu empfehlen, wenn sehr lange Host-Namen von anderen Systemen übernommen, allerdings der Einfachheit halber über kürzere Alias-Namen angesprochen werden sollen.

Betrachtet man das Namensverfahren mittels einer Datei (*Hosts*), so ist erkennbar, dass zwar die Implementierung recht einfach vorgenommen werden kann, die Wartbarkeit dieses Systems sich in größeren Netzwerken jedoch als nahezu unmöglich darstellt. Änderungen in der Netztopologie ziehen immer ein Update der einzelnen lokalen *Hosts*-Dateien nach sich; der Personalaufwand ist dafür unvertretbar hoch. Außerdem ist die Handhabbarkeit einer permanent wachsenden *Hosts*-Datei sicher nicht als vorteilhaft zu beurteilen, zumal diese manuell vom Netzwerkadministrator gepflegt werden muss. Dies ist der Grund, dass heutzutage zur Auflösung von symbolischen Namen fast immer DNS (*Domain Name System*) zum Einsatz kommt (siehe unten).

Namenshierarchie

In der Praxis kommt zur Namensauflösung lediglich ein System infrage, das dezentral organisiert ist. Zentrale Systeme werden aus allen Bereichen eines Netzwerks sternförmig angesprochen und führen zu einer stark erhöhten Netzwerkbelastung. So werden verteilte Namensserver eingerichtet, die für einen fest definierten Netzwerkbereich (Domänen) zuständig sind. Nur für den Datenverkehr zwischen den Domänen erfolgt eine übergreifende Kommunikation.

Die meisten Netzwerke weisen eine hierarchische Struktur auf. Es existieren kleinere Abteilungsnetze, die über Backbones zu Bereichs- oder Unternehmensnetzwerken zusammengefasst werden. Bei einer dezentralen Unternehmensstruktur ergibt sich oft eine weitere Verbindung der Unternehmensnetze untereinander im *Wide Area Network*. Will man ein derartiges Netzwerkverbundsystem mit einer symbolischen Namensadressierung versehen, so sollte diese ebenso einer hierarchischen Struktur entsprechen.

Ein durchdachtes Konzept zur Generierung einer Adresshierarchie für symbolische Namen reflektiert normalerweise die Unternehmensstruktur. Aufgaben im DNS werden auf verschiedene dezentrale Nameserver verteilt, die

ihrerseits einen mehr oder weniger umfangreichen Hierarchiebaum verwalten. Die unterste Hierarchiestufe eines Namensbaums sind die Top Level Domains (TLD). Sie wurden bereits in den ersten Jahren der »Gründung« des Internets festgelegt und in den USA verwaltet. Entgegen den Gepflogenheiten anderer Staaten setzt sich diese unterste Hierarchiestufe in den USA aus organisatorischen Strukturbezeichnungen zusammen (z.B. COM für »Commercial Organizations« oder GOV für »Government Organizations«). Mit zunehmender Internationalisierung des Internets wurde dann ein Länderkürzel eingeführt, das sich weltweit etabliert hat. So erhielt Deutschland beispielsweise das Kürzel DE, Frankreich FR oder das Vereinigte Königreich UK. Mittlerweile wurden auch weitere Top Level Domains eingeführt – nachzulesen unter www.iana.org.

HINWEIS

Die hierarchische Struktur wird unterhalb der Top-Level-Domains weiter fortgesetzt und kann mehrere Zonen umfassen. Jede einzelne Zone wird in der Notation mit einem Dezimalpunkt von anderen Zonen getrennt. Jede Zone betreibt normalerweise einen eigenen *Primary Name Server* und zur

Gewährleistung eines ausfallsicheren Betriebs auch einen *Secondary Name Server*.

Funktionsweise

Gemäß den beschriebenen Namensstrukturen innerhalb der einzelnen Domänen bzw. Zonen lässt sich allerdings keine direkte Adressierung des gewünschten Kommunikationspartners vornehmen. Ebenso wie zur tatsächlich physischen Adressierung über das *Address Resolution Protocol* (ARP) die MAC-Adresse des Systems (bzw. des Netzwerkcontrollers) ermittelt wird, so erfolgt hier auf einer anwendungsnahen Protokollschicht die Umsetzung eines symbolischen Namens auf die IP-Adresse.

Zur Durchführung der Namensauflösung in die IP-Adresse gibt es zwei Möglichkeiten: Der entsprechende Host kennt seinen Nameserver, der ihm Anfragen nach der IP-Adresse des angegebenen Namens beantwortet. Dabei findet eine Client-Server-Kommunikation zwischen dem *resolver client* und dem *name server process* statt. Der Client übermittelt dem Server den Namen, der aufgelöst werden soll, und gibt ferner an, wie der Auflösungsprozess vorzunehmen ist. Wenn der Client vom Server eine *nonrecursive resolution* fordert (auch *iterative resolution* genannt), dann ermittelt der Server lediglich die

Nameserver, die für die Auflösung des angeforderten Namens zuständig sind, und teilt diese dem Client mit. Es ist dann Aufgabe des Clients, durch eine erneute Anfrage an die nun ermittelten tatsächlich zuständigen Nameserver die IP-Adresse des Namens zu erhalten. Sollte der Client allerdings eine *recursive resolution* anfordern, so hat der angesprochene Nameserver für die vollständige Auflösung des Namens zu sorgen, und der Client ist aus der Pflicht. Er erhält dann allein vom kontaktierten Nameserver die gewünschte Auflösungsantwort. Dieses Verfahren ist in der Praxis häufig anzutreffen, da man die einzelnen Clients (es handelt sich oft um kleine, nicht sehr leistungsfähige Systeme) von der Request-Aktivität entlasten will. Zur Reduzierung der Netzwerklast und zur Optimierung der *Response-Zeit* wird vom *Primary Name Server* ein Cache gebildet, der alle bereits von weiteren Nameservern bezogenen Namensauflösungen aufnimmt, diese bei Bedarf erneut hervorholt und an den Anfragenden überträgt. Erst wenn der Cache eine Namensauflösung nicht durchführen kann, wird der Kontakt zu einem weiteren Server hergestellt.

Es gibt unterschiedliche Verfahren zur Auflösung von symbolischen Namen in IP-Adressen. Die beiden heutzutage am häufigsten benutzten Möglichkeiten werden nachfolgend erläutert.

Statische Namensauflösung

In der Vergangenheit erfolgte die Namensauflösung mithilfe einer speziellen Datei, die auf jedem System verfügbar sein musste. Diese Datei mit dem Namen *hosts* beinhaltet sämtliche Zuordnungen von Host-Namen zu den jeweiligen IP-Adressen in Form einer statischen Datei. Im Extremfall sind in einer solchen Datei somit sämtliche Angaben zu allen Hosts eines Netzwerks abgelegt.

Speziell in den Anfangszeiten des Internets gab es tatsächlich eine solche Hosts-Datei, die den Namen »hosts.txt« trug und alle Host-Namen des Internets samt IP-Adressen enthielt. Jeder Systemverwalter am Netz musste diese Datei regelmäßig herunterladen, um so auf alle Rechnernamen zugreifen zu können. Durch das rasche Wachstum des Internets war diese Möglichkeit bald überholt. Zudem mussten alle Rechnernamen eindeutig sein, was bei der riesengroßen Anzahl von Rechnern nicht einfach war.

HINWEIS

Der Einsatz einer Hosts-Datei kann bis zu einer gewissen Anzahl von Endgeräten durchaus Sinn ergeben. In größeren

Netzwerken sollten jedoch unbedingt DNS- oder WINS-Server anstelle der Hosts-Datei eingesetzt werden.

Windows-Hosts-Datei

Auf einem Windows-System befindet sich die Datei *hosts* im Verzeichnis `|%systemroot%|system32\drivers\etc`. Der Inhalt der Dateien ist dabei vergleichbar mit anderen Systemen (UNIX, Linux usw.), wobei nachfolgend einmal beispielhaft der Inhalt einer Host-Datei eines Windows-Systems dargestellt wird.

```
#

# HOSTS-Datei für System 190.168.66.12

#

# IP-Adresse           Host-Name           Kommenta
190.168.66.11          gerd                # PC Ger
190.168.66.44          eva                 # PC Eva
190.168.69.12          dirk pcdirk         # PC Dir
```

```
190.168.70.25      willi      # AXP-Re
```

Am Anfang der Datei stehen einige Kommentarzeilen, die grundsätzlich alle mit dem vorangestellten Kommentarzeichen (#) beginnen. Die entscheidenden Einträge in dieser Beispieldatei befinden sich am Ende. Dort erfolgt die Zuordnung von Host-Namen zu den jeweiligen IP-Adressen. So wird damit beispielsweise dem Host mit dem Namen *dirk* die IP-Adresse 190.168.69.12 zugeordnet. Auffallend ist, dass dem Eintrag für die IP-Adresse 190.168.69.12 zwei Host-Namen zugeordnet sind; zum einen *dirk* und als zweiter Name *pcdirk*. Auch dies ist in einer Hosts-Datei möglich: Mehrfachzuordnung von Namen zu einer IP-Adresse.

Möchte beispielsweise ein System (z.B. Rechner) mit einem der hier aufgeführten Netzwerkknoten Kontakt aufnehmen (z.B. zwecks einer TELNET-Sitzung), so braucht er das nicht mehr durch die Eingabe des Befehls *telnet 190.168.69.12* zu tun, sondern er kann Folgendes eingeben: *telnet dirk*. Der IP-Prozess (*Resolver*) sieht unmittelbar nach Aktivierung des TELNET-Prozesses in der Hosts-Datei nach und versucht, den ihm übergebenen Namen (*dirk*) anhand der vorliegenden Datei aufzulösen. Gelingt ihm das, d.h., es existiert ein Hosts-Eintrag

für das gewünschte Kommunikationsziel, so wird über die ermittelte IP-Adresse eine TELNET-Sitzung etabliert.

In allen Fällen erfolgt eine Namensauflösung der jeweiligen Host-Namen und eine Zuweisung an die in der Hosts-Datei zugeordnete IP-Adresse. In diesem Zusammenhang ist jedoch unbedingt zu beachten, dass die Namensauflösung immer explizit an das Endgerät (Rechner o.Ä.) gebunden ist, auf dem die Hosts-Datei generiert bzw. gepflegt wird.

HINWEIS

Mit Einsatz einer Hosts-Datei ist es notwendig, dass auf jedem Host (Rechner im Netzwerk) eine entsprechende Datei gepflegt wird. Dadurch wird ein solches Verfahren auch sehr schnell unübersichtlich und es gilt die Empfehlung, stattdessen Nameserver-Verfahren wie WINS und DNS einzusetzen (siehe unten).

UNIX-Hosts-Datei

Unter UNIX (Linux usw.) ist die Hosts-Datei standardmäßig im Verzeichnis */etc* abgelegt, trägt dort jedoch den gleichen Namen (*hosts*) wie unter Windows. Im nachfolgenden Beispiel soll anhand des Betriebssystems Linux dargestellt werden, auf welche Art und Weise eine Auflösung von Namen mithilfe

einfacher Einträge in einer Hosts-Datei auf einem UNIX-System erfolgen kann. Der Inhalt einer solchen Datei kann sich beispielsweise wie folgt darstellen:

```
# hosts      This file describes a number of hosts and
#
#            mappings for the TCP/IP subsystem. It
#
#            used at boot time, when no name server
#
#            On small systems, this file can be used
#
#            "named" name server.
#
# Syntax:
#
#
# IP-Address Full-Qualified-Hostname Short-Hostname
#
127.0.0.1    localhost
```

```
192.168.0.55      dirk dirk.dilaro.de  
  
192.168.0.56      hans  
  
192.168.0.57      SERVER01 s1
```

Genau wie auf einem Windows-System erfolgt auch unter UNIX eine Zuweisung von Namen zu IP-Adressen. Auffallend ist an diesem Beispiel wiederum die Möglichkeit der Mehrfachzuordnung, wobei beispielsweise der IP-Adresse 192.168.0.55 die beiden Host-Namen *dirk* und *dirk.dilaro.de* (kompletter Name mit Domäne) zugewiesen werden.

Mit der Zuordnung von Host-Namen zu IP-Adressen können dann beispielsweise Anweisungen der folgenden Art eingesetzt werden:

```
ping dirk  
  
ftp dirk.dilaro.de  
  
ssh hans
```


HINWEIS

Nähere Angaben zu den hier erwähnten Dienstprogrammen wie PING, SSH und FTP enthalten die nachfolgenden Kapitel dieses Buches.

Merkmale einer Hosts-Datei

Die wesentlichen Punkte bei der Behandlung einer Hosts-Datei sind nachfolgend noch einmal zusammengefasst:

- Der Name der Hosts-Datei lautet auf allen Systemen immer *hosts*.
- Der Einsatz einer Hosts-Datei sollte in Netzwerken mit mehr als 20 Endgeräten (Knoten) aus Gründen der Übersichtlichkeit und Pflegbarkeit nicht mehr zum Einsatz kommen.
- In einer Hosts-Datei erfolgt pro Zeile jeweils ein Eintrag.
- Die einzelnen Zeilen einer Host-Datei müssen jeweils mit Carriage Return (CR; wird durch Betätigung der ENTER- oder RETURN-Taste erreicht) abgeschlossen werden.
- Jede Zeile beginnt mit einer IP-Adresse, gefolgt von dem Host-Namen.

- Zwischen der IP-Adresse und dem Host-Namen muss mindestens ein Leerzeichen stehen.
- Ein Host-Name darf maximal 255 Zeichen lang sein, wobei alphanumerische Zeichen und einige Sonderzeichen eingesetzt werden dürfen. Ein Host-Name, der nur aus Ziffern besteht, ist nicht zulässig.

HINWEIS

Der Unterstrich ist zwar als gültiges Zeichen eines Host-Namens zulässig, sollte aber vermieden werden, da dies unter Umständen zu Problemen führen kann.

- Die Groß-/Kleinschreibung bei Host-Namen wird auf UNIX-Systemen beachtet.
- Häufig benutzte Hosts sollten oben in der Datei stehen, da diese von oben nach unten gelesen wird.
- Pro IP-Adresse können mehrere Host-Namen zugeordnet werden, wobei mehrere Namenseinträge mindestens durch ein Leerzeichen getrennt sein müssen.
- Kommentarzeilen werden in einer Hosts-Datei immer mit dem Zeichen »#« eingeleitet.
- Die Hosts-Datei wird unter UNIX im Verzeichnis */etc* und auf einem Windows-System im Verzeichnis

|%systemroot\system32\drivers\etc abgelegt.

HINWEIS

Neben der Hosts-Datei wird in der Literatur hin und wieder auch die Datei *lmhosts* erwähnt. Dies ist eine Datei, in der eine Zuordnung von NetBIOS-Namen zu IP-Adressen abgelegt werden kann.

Da selbst Microsoft auf seiner Homepage den Einsatz seines proprietären Dienstes zur Namensauflösung WINS nicht mehr empfiehlt und stattdessen auf die Einrichtung von DNS verweist, geht dieses Buch nicht weiter auf WINS ein.

Dynamische Namensauflösung

Die Probleme, die sich durch die Verwaltung von Internet-Hosts mittels einer Hosts-Datei ergaben, wurden bereits an anderer Stelle beschrieben. Als Folge dessen wurde das *DNS (Domain Name System* oder auch oft *Domain Name Service*) entwickelt, dessen wesentliche Aufgaben und Funktionen im nachfolgenden Abschnitt erläutert werden.

Das *Domain Name System* (DNS) stellt heutzutage das wichtigste System zur Verwaltung und Zuweisung von IP-Adressen zu Namen dar (Namensauflösung). Dies sicherlich

nicht zuletzt aus der Tatsache heraus, dass es sich um ein standardisiertes Verfahren handelt, das in diversen RFCs immer wieder verfeinert und optimiert worden ist. So ist das DNS einer der wichtigsten Internetdienste, der in dem nahezu undurchdringlichen »Internetdschungel« von IP-Adressen die einzige Orientierungshilfe darstellt.

HINWEIS

Das DNS hat nichts mit Routing zu tun, denn es ist daraus grundsätzlich nicht abzuleiten, wo die einzelnen Endgeräte platziert sind. Es geht beim DNS ausschließlich um die Verwaltung von Namen und den zugehörigen IP-Adressen.

Aufgaben und Funktionen

Das DNS ist nicht nur für die Rechneradressierung über »sprechende« Namen zuständig, sondern erfüllt insgesamt einige wesentliche Aufgaben:

- Umwandlung »sprechender« Namen in IP-Adressen
- Umwandlung von IP-Adressen in »sprechende« Namen (Reverse Translation)
- Verwaltung einer oder mehrerer Domänen und ihrer Namensdatenbasis

- Verwaltung von MX-Records – Bereitstellung der Adressinformationen zuständiger Mail-Server für die jeweiligen Zieldomänen

Die Einrichtung von DNS-Servern ist nicht nur für die externe Kommunikation im Internet geboten, sondern sie stellt auch die Grundlage der Adressierung in einem Intranet dar. Jeder Anwender, der auf Informationen eines Intranetservers (unternehmensinterner Webserver) zugreifen will, benötigt den »Kontakt« zu einem internen DNS-Server. Kein Systemverwalter wird ernsthaft auf die Idee kommen, die interne Adressierung eines TCP/IP-Netzwerks auf die Verwendung von IP-Adressen zu beschränken. Die Vielzahl unterschiedlicher Serversysteme in einem Unternehmen würde zu einer »Orientierungslosigkeit« führen, da die verwendeten IP-Adressen keinerlei Rückschlüsse auf den Server zulassen. Die Adressierung eines Servers mit Namen »SAP-FINANZ« ist sicher besser nachzuvollziehen als die Angabe von »192.168.22.1«. Die Verwendung einer Hosts-Datei ist keine vernünftige Alternative, da diese Dateien auf jedem Client-System manuell gepflegt werden müssten.

Auflösung von Namen

Die Idee, die dem DNS zugrunde liegt, ist zunächst einmal sehr einfach. Da das Internet zu groß ist, um seine Namen und IP-

Adressen zentral zu verwalten, muss eine Aufteilung in dezentrale Teilbereiche vorgenommen werden. Diese Teilbereiche werden dabei als *Domain* (Domäne) bezeichnet. Das DNS basiert auf der Idee von Domains, also unterteilten Bereichen, die organisatorisch für die Verwaltung der Host-Namen zuständig sind.

HINWEIS

DNS-Domänen besitzen keine inhaltliche Gemeinsamkeit mit Domänen, die beispielsweise in einem Windows-System zum Einsatz kommen.

Allgemein ausgedrückt handelt es sich bei dem Domain Name System (DNS) um eine verteilte hierarchische Datenbank, in der verschiedenste Informationen über die Endgeräte eines IP-Netzwerks gespeichert werden. So erfolgt per DNS beispielsweise die Auflösung von Rechnernamen, die besser zu merken sind als Zahlen, in die zugehörigen IP-Adressen, die wiederum für die weitere Verarbeitung benötigt werden. So ist es beispielsweise viel einfacher, statt einer entsprechenden IP-Adresse Folgendes als Webadresse (URL) einzugeben:

<http://www.dilaro.de>

Internetadressen in der dargestellten Form kennen sicherlich die meisten, obwohl es sich streng genommen überhaupt nicht um Adressen handelt. Denn wie an anderer Stelle dieses Buches erläutert, stellen sich »echte« Internetadressen in folgender Form dar:

212.227.109.208

Da es naturgemäß schwieriger ist, sich Zahlenfolgen zu merken, wurde eine Systematik entwickelt, die es ermöglicht, einer Internetadresse einen Namen zuzuweisen. So verbirgt sich in dem obigen Beispiel hinter [www.dilaro.de](http://212.227.109.208) nichts anderes als die IP-Adresse 212.227.109.208. Die entsprechende Eingabe in der Adresszeile eines Webbrowsers könnte sich demnach auch wie folgt darstellen:

<http://212.227.109.208>

Bei der Frage nach der Verwaltung entsprechender Internetadressen und der zugeordneten Namen, wie beispielsweise www.dilaro.de, ergibt sich automatisch die Verbindung zum DNS. Das DNS ermöglicht, dass für die einzelnen Endgeräte eines IP-Netzwerks symbolische Namen (Host-Namen) vergeben werden. Auf diese Art und Weise ergibt sich der Zugriff auf ein Gerät nicht über die IP-Adresse, sondern über den zugewiesenen Namen.

Damit das DNS eine Namensauflösung (so wird die Konvertierung symbolischer Namen in IP-Adressen bezeichnet) durchführen kann, werden sogenannte *Nameserver* benötigt, wobei alle Adressen in einer großen hierarchischen, weltweit verteilten Datenbank abgelegt sind. Wenn ein Nameserver eine Adresse nicht umsetzen kann, gibt er die Anfrage an einen ihm übergeordneten Nameserver weiter. Durch dieses Verfahren sind die Adressen leichter zu merken (aussagekräftige Namen statt Zahlenkolonnen) und einfacher zu verwalten. Sobald ein Rechner eine andere IP-Adresse bekommt, muss nur der Eintrag auf dem Nameserver geändert werden. Der Name, unter dem er zu erreichen ist, bleibt gleich.

Organisationen mit eigenen Domains müssen einen Nameserver unterhalten, der die Namen in Internetadressen umsetzt. Dies ist notwendig, damit alle Rechner des Internets

die symbolischen Rechnernamen einer Organisation in Internetadressen umsetzen lassen können, ohne die eine Kommunikation nicht möglich ist. Dabei sind Nameserver zunächst einmal nichts anderes als Rechner, die Datenbanken über Namen und die dazugehörigen IP-Adressen der betreffenden Domain verwalten. Ist ein Nameserver ein TLD-Server, so verfügt er über sämtliche Adressen aller anderen Top-Level-Nameserver.

Im Gegensatz zu den Nameservern werden die Programme, die die Informationen eines Nameservers nutzen, mit dem Begriff *Resolver* (Auflöser) umschrieben. Es handelt sich dabei in der Regel um Programme, die in die Anwendungen eingebunden werden und so die Funktionen nutzen (z.B. TELNET). Die Resolver-Software stellt Anfragen an den lokalen Nameserver, um Namen in IP-Adressen zu übersetzen. Entweder sind die dazu benötigten Informationen im lokalen Nameserver vorhanden oder dieser muss seinerseits andere Nameserver konsultieren, um die gewünschte Umsetzung zu leisten.

DNS-Struktur

Bis zum Jahr 1984 wurde die Zuordnung von Namen zu IP-Adressen von einer zentralen Stelle gepflegt, und zwar für alle

weltweit vergebenen (offiziellen) IP-Adressen. Zuständig war das *Network Information Center* (NIC) in den USA. Da dieses Verfahren mit dem Anwachsen des Internets zu unübersichtlich wurde, wurde das DNS-Verfahren eingeführt. In Zeiten täglich zunehmender Internetnutzung spielt der Einsatz eines Namensdienstes wie DNS eine immer wichtigere Rolle. Jedoch auch im lokalen Netzwerk eines Unternehmens (LAN, WAN) ist das DNS ein unverzichtbarer Dienst. So werden beim DNS alle Namen (und deren IP-Adressen) in einer Domäne verwaltet, wobei die hierarchische Organisation einer DNS-Datenbank mit der Verzeichnisstruktur eines Dateisystems vergleichbar ist: Beide haben eine umgekehrte baumartige Struktur, bei der die Wurzel oben und die Zweige unten stehen. Namen, die alle Knoten zwischen dem Wurzel- und dem Endknoten enthalten, werden als *vollqualifizierte Domänennamen* (FQDN, *Fully Qualified Domain Name*) bezeichnet.

HINWEIS

Die gesamte Struktur einer hierarchisch organisierten DNS-Datenbank wird auch als *Domain Name Space* (Domänen-Namensraum) bezeichnet. Der Wurzelknoten (Ursprung des Baums) im Domain Name Space wird als *Root Domain* (Wurzel) oder Hauptdomäne bezeichnet. Domänen (Knoten), die direkt

dem Wurzelverzeichnis untergeordnet sind, werden als *Top Level Domains* (Top-Level-Domänen) bezeichnet. Jeder Top Level Domain (TLD) können wiederum weitere Domänen (Subdomains) untergeordnet sein.

Die TLD bildet stets die oberste Ebene einer Domäne und wird auch als Wurzelverzeichnis (Root) bezeichnet.

Der gesamte Domänen-Namensraum wird in Zonen unterteilt; diese Zonen (*Subdomains*) wiederum sind Untermengen des DNS-Baums. Die Forderung nach Eindeutigkeit wird bei DNS auf den Bereich einer *Subdomain* beschränkt. So können in unterschiedlichen Subdomains durchaus gleiche Bezeichnungen (Namen) vergeben werden. Um eine große Anzahl von Namen zu verwalten und gleichzeitig den Organisationen Freiheit in der Namenswahl zu ermöglichen, wurde eine hierarchische Struktur entworfen.

Jeder Knoten im DNS-Baum trägt einen Namen, der bis zu 63 Zeichen lang sein kann. Die vollqualifizierten Domännennamen (*Fully Qualified Domain Name* = FQDN) beginnen immer mit dem Zielnamen und gehen zurück zur Root, wobei die einzelnen Knoten jeweils durch Punkte getrennt werden. Das Wurzelverzeichnis wird (laut Standard) durch einen nachgestellten Punkt gekennzeichnet, der jedoch in der Regel

nicht mit angegeben wird. Ein vollständiger FQDN könnte somit beispielsweise wie folgt aussehen: [www.dilaro.doc.de](#), dabei wäre die Schreibweise [www.dilaro.doc.de](#) jedoch die richtige Schreibweise (mit Punkt am Ende). In diesem Beispiel ist *de* die Top Level Domain und jedes weitere Level definiert eine Subdomain (z.B. *doc*) bzw. einen Netzknoten (z.B. *dilaro*).

Jede Zone eines DNS-Baums muss mindestens über einen primären Nameserver verfügen. Ein zweiter (sekundärer) Nameserver sollte aus Gründen des Ausfallschutzes eingerichtet werden. Auf einem primären Nameserver werden in einem solchen Fall dann sekundäre Zonen eingerichtet, die als primäre Zonen auf dem sekundären Nameserver existieren. Umgekehrt werden die primären Zonen des sekundären Nameservers als sekundäre Zonen auf dem primären Nameserver eingerichtet. Die einzelnen Nameserver müssen die jeweiligen Zonendaten untereinander austauschen.

Den größten Domain Name Space stellt das Internet dar. Dort ist eine Reihe sogenannter Root-Nameserver in Betrieb (z.B. im NFSnet, MILNET, SPAN), die für die Top Level Domains verantwortlich sind. Die weitere Verwaltung der Nameserver in den unteren Zonen wird delegiert. In der zweiten und dritten Ebene sind hier juristische Personen wie Behörden und Universitäten mit dem Betrieb der Nameserver betraut. Diese

sind jeweils für die Verwaltung ihrer Zweige im DNS-Baum verantwortlich.

DNS-Anfragen

Nach der Erläuterung des Grundprinzips stellt sich die Frage, wie eine entsprechende DNS-Namensauflösung in der Praxis abläuft. Wenn beispielsweise ein Endgerät ein Datenpaket an ein anderes Endgerät versendet, muss es zuerst herausfinden (lassen), welche IP-Adresse das andere Endgerät hat. Erst dann kann die eigentliche Wegewahl (Routing) beginnen, denn die Tabellen zur Wegewahl basieren allesamt auf IP-Adressen.

Um die Auflösung eines Host-Namens zu bewerkstelligen, wird ein entsprechender Nameserver (DNS-Server) eingesetzt. Es wurde bereits erwähnt, dass ein Anwendungsprogramm, das auf die Daten eines DNS-Systems zugreift, sich dabei den Einsatz des sogenannten *Resolver* zunutze macht.

HINWEIS

Der Vorgang, einen Nameserver abzufragen, wird als *Nameserver-Lookup*, als *DNS-Lookup* oder auch als *Forward DNS-Lookup* bezeichnet.

Der *Resolver* ist die Software, die auf einem DNS-Client für den Zugriff auf einen Nameserver sorgt. So erzeugt er auf einem DNS-Client (Host) eine DNS-Anfrage und sendet diese an den Nameserver. Der Nameserver sendet die Antwort (Auflösung des Namens) zurück an den Resolver, der die Daten verarbeitet und an die Anwendung liefert. Der Nameserver liefert die IP-Adresse zu einem vom Resolver übermittelten Namen, sofern sie ihm bekannt ist. Kennt er die IP-Adresse nicht bzw. kann er den Namen nicht auflösen, wird die Anfrage zum übergeordneten Nameserver weitergeleitet. Wenn nötig, wird die Anfrage zur obersten Ebene von Nameservern weitergeleitet, bis die Anfrage aufgelöst werden kann. Die Ergebnisse der letzten Namensauflösung speichert der Nameserver in einem lokalen Zwischenspeicher. Dies verkürzt die Anfrage, falls die IP-Adresse zur Namensauflösung sonst mehrere Nameserver durchlaufen müsste. Auch sind dem Nameserver hierdurch Adressen anderer Nameserver in anderen Zonen des DNS-Baums bekannt. Dadurch kann die Anfrage direkt an einen Nameserver in einer anderen Zone geleitet werden, ohne den Umweg über die oberste Ebene von Nameservern zu gehen.

Der Ablauf einer Namensauflösung ist grundsätzlich ein sehr aufwendiges Verfahren, das hier jedoch in vereinfachter

Form dargestellt werden soll, um den generellen Ablauf zu verdeutlichen:

- Sobald eine Namensanfrage erfolgt (z.B. durch eine Anweisung der Form *telnet server01*), wird als Erstes überprüft, ob der lokale Name des betreffenden Endgeräts mit dem Host-Namen (hier: *SERVER01*) identisch ist.
- Ist dies nicht der Fall, wird in der lokalen Hosts-Datei ein Eintrag für den Host (SERVER01) gesucht. Ist ein entsprechender Eintrag vorhanden, wird die zugehörige IP-Adresse an die Anwendung (hier: TELNET) zurückübermittelt.
- Kann der Host-Name in der lokalen Hosts-Datei nicht aufgelöst werden, wird als Nächstes ein DNS-Server (Nameserver) angesprochen. Dies setzt natürlich voraus, dass das anfragende Endgerät auch als DNS-Client konfiguriert worden ist.
- Der DNS-Server (primärer DNS-Server) sucht den übermittelten Host-Namen in seiner Datenbank.
- Findet er den Namen in seiner Datenbank, sucht er die dazugehörige IP-Adresse heraus und übermittelt diese an den anfragenden Host.
- Kann der (primäre) DNS-Server den Host-Namen nicht auflösen, gibt er die Anfrage an einen übergeordneten DNS-

Server weiter. Dies kann bis zu einem DNS-Server auf oberster Ebene führen.

- Besteht keine Möglichkeit, den angeforderten Namen aufzulösen, erscheint eine entsprechende Fehlermeldung (unbekannter Host, Zeitüberschreitung o.Ä.).

Umgekehrte Auflösung

Nameserver werden eingesetzt, um vorgegebene Namen in IP-Adressen umzuwandeln bzw. diesen Namen die entsprechenden IP-Adressen zuzuordnen. In der Praxis wird aber auch hin und wieder der umgekehrte Fall, also die Zuordnung eines Host-Namens zu einer vorgegebenen IP-Adresse, vorkommen. Für diesen speziellen Anwendungsfall der umgekehrten Auflösung steht im Internet die Domäne mit dem Namen ARPA zur Verfügung, die als *in-addr.arpa.* konfiguriert wird. Das Wesentliche bei der umgekehrten Zuordnung von IP-Adressen zu Namen ist, dass den einzelnen Knoten jeweils die IP-Adresse zugewiesen wird.

Die Domäne *in-addr.arpa.* kann insgesamt bis zu 256 Subdomains verwalten. Diese entsprechen dem ersten Oktett der IP-Adresse. Diese Subdomains (des ersten Oktetts) können jeweils 256 Subdomains haben, die dem zweiten Oktett der IP-Adresse entsprechen. Diese wiederum können 256 Subdomains

entsprechend dem dritten Oktett der IP-Adresse besitzen. Die letzte Subdomain kann 256 Datensätze entsprechend dem letzten Oktett der IP-Adresse besitzen. Somit entspricht der Wert eines Datensatzes aus dem vierten Oktett der IP-Adresse dem FQDN der IP-Adresse.

HINWEIS

Der umgekehrte Vorgang der Auflösung einer IP-Adresse in einen Host-Namen wird auch als *Reverse DNS-Lookup* bezeichnet. Zum Einsatz kommt dabei eine spezielle Anweisung mit dem Namen NSLOOKUP.

Standard Resource Records

Die Basisdatei eines Nameservers, die auf UNIX-Systemen mit *named.boot* bezeichnet wird und zumeist im Verzeichnis */etc* zu finden ist, zeichnet für die Steuerung der DNS-Ressourcendateien verantwortlich. Windows-Systeme beziehen ihre Konfiguration in der Regel aus der Registry-Datenbank.

Das Satzformat der DNS-Ressourcendateien, die den relevanten Datenbestand aufnehmen sollen, ist festgelegt und besitzt einen besonderen Aufbau. Sie sind dem RFC 1035 entnommen:

- *NAME*

Name des Eigners dieses Resource Record (Host-Name)

- *TYPE*

Ein zwei Bytes umfassendes Feld mit folgendem Inhalt:

- A: IP-Adresse
- NS: autorisierter Nameserver
- MD: alte Angabe; neu MX
- MF: alte Angabe; neu MX

- *CNAME*

Angabe eines Alias-Namens

- *SOA*

Gibt den Beginn einer Zone an (Start Of Authority)

- *MB, MG, MR, NULL*

Experimentell

- *WKS*

Angabe von Diensten (*well known services*)

- *PTR*

Angabe eines Zeigers auf eine Domäne

- *HINFO*

Textinformation zur Beschreibung eines Hosts

- *MINFO*

Mailbox- oder Mailing-List-Informationen

- *MX*

Verteiler für Mail-Dienste

- *TXT*

Zeichenketten

- *CLASS*

Ein zwei Bytes umfassendes Feld mit folgendem Inhalt:

- IN: Internet-Protokollfamilie TCP/IP
- CS: alte Angabe; wird nicht mehr verwendet
- CH: die CHAOS-Klasse; wird nicht verwendet
- HS: Hesiod; wird nicht verwendet

- *TTL*

Ein vier Bytes umfassendes Integer-Feld. Es gibt den Zeitraum (in Sekunden) für beim Nameserver erneut angeforderte DNS-Daten an. Wird hier kein Wert angegeben, so wird der entsprechende Eintrag im SOA-RR verwendet. Bei der Wahl dieses Wertes sollte man sorgfältig vorgehen. Bei einem zu niedrigen Wert wird nämlich die Leistungsfähigkeit des Servers durch häufiges Anfordern reduziert; hingegen wird die Reaktionszeit des Servers negativ beeinflusst, wenn der TTL-Wert (*Time to Live*) zu hoch angesetzt ist.

- *RDLENGTH*

Ein zwei Bytes umfassendes Feld, das die Länge des folgenden RDATA-Felds angibt

- *RDATA*

In Abhängigkeit des TYPE- und CLASS-Formats werden hier Daten mit variabler Länge zur Beschreibung der Ressource

angegeben.

DNS-Message

Die DNS-Kommunikation wird durch das Wechselspiel zwischen *DNS-Request (Resolver)* und *DNS-Response (Server)* charakterisiert. Die Datenpakete sind im *DNS Message Format* beschrieben und umfassen die Abschnitte *Header*, *Question*, *Answer*, *Authority* und *Additional Information Sections*.

Wie bereits bei zahlreichen höheren Protokollen bilden auch hier ein *MAC-Header*, ein *IP-Header* und schließlich der *DNS-Header* die Grundlage für die eigentliche DNS-Message. Der *DNS-Header* umfasst die folgenden Felder:

- *Identifikation (16)*

Erforderlich zur Antwort-Frage-Zuordnung.

- *Parameter (16)*

Verschiedene Teilparameter werden hinterlegt:

- operation type: query/response
- query type: standard/inverse/obsolete
- answer authoritative
- message is truncated
- recursion is desired
- recursion is available
- number of questions

- *Anzahl Antworten* (16)
- *Anzahl Quellen* (16)
- *Anzahl Zusätze* (16)

Die Felder der DNS-Message sind hauptsächlich für die Formulierung der *questions* und den Empfang der *answers* zuständig:

- *Anfragen-Information (v)*:
 - Domain-Name der Anfrage
 - Art der Anfrage
 - Klasse der Anfrage (Protokollgruppe)
- *Antwort-Information (v)*:
 - resource domain name
 - type
 - class
 - time to live
 - resource data length
 - resource data
- *Quellen-Information (v)*:

Enthält den Namen des Servers, der letztlich die Auflösungsinformationen geliefert hat (möglicherweise über die Kontaktaufnahme mehrerer vorgelagerter Server).
- *Zusatz-Information (v)*

Dynamic DNS (DDNS)

Unter *Dynamic DNS* versteht man die automatische Anpassung der Namensauflösung bei Änderung der IP-Adresse eines Endgeräts (z.B. Rechner). So stellt DDNS die automatische Anpassung der Namensauflösung bei der Änderung der IP-Adresse eines IP-Knotens sicher. Davon sind sowohl der *Forward*- (Auflösung vom Namen zur IP-Adresse) als auch der *Reverse*-Prozess (Auflösung von IP-Adresse zum Namen) betroffen. Der wesentliche Unterschied zum »klassischen DNS« besteht darin, dass die DNS-Registrierung der IP-Adresse eines Clients automatisch erfolgt und somit eine explizite Pflege einer statischen DNS-Datenbasis nicht mehr erforderlich ist. Die Registrierung wird von einem DHCP-Server nach folgendem Verfahren vorgenommen:

- Der Client sendet eine IP-Adressanforderung an den DHCP-Server.
- Der DHCP-Server empfängt die Anforderung, teilt eine IP-Adresse zu und legt die Gültigkeitsdauer fest (*Lease*).
- Die IP-Adresse des Clients wird vom DHCP-Server als A-Adresse beim DNS-Server registriert.
- Der DHCP-Server registriert beim DNS-Server den PTR-Eintrag zur Reverse-Zone der Domäne.

HINWEIS

Windows-Systeme können ihre IP-Adresse auch direkt beim DNS-Server registrieren; andere Clients werden über DHCP-Server registriert. Außerhalb der Windows-Betriebssysteme wird DDNS in der UNIX-Welt auch von BIND ab Version 8.1.2 unterstützt. Die RFCs 2136 und 2137 beschreiben die als DDNS definierten dynamischen DNS-Aktualisierungen.

Zusammenspiel von DNS und Active Directory

Verzeichnisdienste repräsentieren umfangreiche Datensammlungen, die möglichst viele und unterschiedliche Unternehmensressourcen in einer einzigen Datenbank zusammenführen. Eine solche Verzeichnisdatenbank wird zur Adressierung, Dokumentation und Inventarisierung eingesetzt und liefert dem Unternehmen eine solide Planungs- und Administrationsgrundlage.

Active Directory ist ein solcher Verzeichnisdienst und stellt eher eine organisatorische Kommunikationsstruktur dar als ein Kommunikationsprotokoll. Auch wenn es unterschiedliche Verzeichnisdienste gibt, haben sich die *Active Directory Services* (ADS) aus dem Hause Microsoft zumindest auf allen Windows-Plattformen durchgesetzt; aus diesem Grund stehen diese

Dienste im Mittelpunkt der nachfolgenden Betrachtung. Dabei ist von entscheidender Bedeutung, dass bei einer Planung und Einführung der DNS-Integration in *Active Directory Services* (ADS) besondere Aufmerksamkeit gewidmet wird.

Mit *Active Directory Services* wird eine äußerst komplexe und vor allem global gültige Unternehmensdatenbank betrieben, die Informationen zu Benutzern und Ressourcen enthält sowie ihre Verknüpfungen und Freigaben. Mit ihr soll

- eine einheitliche Anmeldung ermöglicht,
- die Benutzer- und Ressourcenadministration vereinheitlicht und
- das Unternehmen hinsichtlich Netzwerken, Benutzern und sonstigen Ressourcen abgebildet werden.

Um diese Ziele zu erreichen, ist eine sehr gründliche und zeitlich aufwendige Projektphase erforderlich, denn die unterschiedlichen Implikationen und Abhängigkeiten existierender Strukturen müssen konsequent ausnahmslos in die neue Active-Directory-Struktur überführt werden.

Directory Service (ADS)

Innerhalb einer ADS-Struktur werden Objekte und ihre Eigenschaften in sogenannte »Schemata« gespeichert. Dem

Objekt »Benutzer« ist beispielsweise die Eigenschaft »Vorname« oder »Beschreibung« zugeordnet, wobei diese Eigenschaften natürlich mehrfach in unterschiedlichen Objekten auftreten können. Objekt-Container werden als Organisationseinheit bezeichnet und stellen den geografischen Bezug zur jeweiligen Domäne her.

Die ADS-Domänenstruktur baut auf einem streng hierarchischen Konzept auf, in dem Domänen und untergeordnete Subdomänen einem umfassenden *Namespace* zugewiesen werden. Dieser repräsentiert eine DNS-Namensstruktur, bestehend aus einem oder mehreren Namespace(s), in dem/denen die verschiedenen Domänen zusammengefasst sind.

Die *Active Directory Services* lassen sich in ihren Strukturen auf einer Vier-Komponenten-Architektur abbilden:

- *Datenmodell*

Als Grundlage dient der X.500-Verzeichnisdienst sowie das *Lightweight Directory Access Protocol* (LDAP). Active Directory bedient sich einer streng hierarchischen Struktur von Ressourcen, Diensten und anderer Objekte.

- *Schema*

In einem Schema wird beschrieben, welche Objekttypen im Directory gespeichert werden. Beispiel: Es werden drei Objektklassen definiert: Server, Organisationseinheiten (OU) und Benutzer. Diesen Klassen werden dann Objekte mit bestimmten Attributen einzelner Werte zugeordnet, wie beispielsweise der Wert 10.1.1.1 für das Attribut *IP-Adresse* und das Objekt *Printserver*. Schemata lassen sich natürlich jederzeit an neue Gegebenheiten anpassen, unterliegen dabei allerdings restriktiven Sicherheitsvorkehrungen hinsichtlich ihrer Administrationsrechte.

- *Sicherheit*

ADS verwendet Kerberos zur Implementierung von Sicherheitsmechanismen. Dies ist insbesondere für die Authentifizierung beliebiger Ressourcen, Benutzer und Dienste relevant.

- *Administration*

Die Active Directory Services ermöglichen eine bedarfsgerechte Möglichkeit zur Verwaltung mittels einer angepassten Rechtevergabe durch den Systemverwalter. Die Vergabe globaler Administrationsrechte für eine ganze Domäne, wie es beim herkömmlichen Windows-Domänenmodell üblich war, gehört der Vergangenheit an.

Die wichtigsten Komponenten eines *Active Directory Service* sind:

- *Namespace*
DNS-Namensstruktur; sie umfasst die Root-DNS-Domäne samt ihren delegierten Sub-Domänen
- *Domäne*
Ansammlung von Ressourcen, die einen gemeinsamen Namespace, ein gemeinsames Schema, eine gemeinsame Konfiguration und einen globalen Katalog (GC) besitzen
- *Domänenstruktur*
Ansammlung von Domänen mit unterschiedlicher Verwendung der SubDomänen
- *Gesamtstruktur*
Mehrere Domänen verfügen über verschiedene Namespaces, teilen sich aber dasselbe Schema, dieselbe Konfiguration und denselben GC.
- *Organisationseinheiten*
Ein Container für Objekte innerhalb einer Domäne
- *Objekte*
Ressource, Dienst oder Benutzerkonto innerhalb der ADS. Sie besitzen Attribute und werden mit Werten versehen, um eine eindeutige Identifizierung zu ermöglichen.
- *Schemata*
Beschreibung der Objekte und deren Eigenschaften innerhalb der ADS
- *Standorte*

Gruppe von IP-Subnetzen

Integration

Mit der Einführung der *Active Directory Services* (ab Windows 2000 Server) erhält DNS nun auch für Windows-Plattformen eine besonders wichtige Bedeutung. Während die Namensauflösung im Windows-Netzwerk früher durch den WINS-Dienst und entsprechende WINS-Server übernommen wurde, erfolgt dies nunmehr über DNS. Um sich beispielsweise an einem Windows-Domaincontroller anzumelden, erfolgt die Suche und Identifikation über DNS; insbesondere erfolgt die Kontaktaufnahme mit einem für die Domäne zuständigen DNS-Server, sofern auf einem Domaincontroller ADS installiert wird. Bleibt die Kontaktaufnahme erfolglos, so initiiert die Installationsroutine für ADS die Einrichtung eines DNS-Servers.

Voraussetzung für einen reibungslosen Einsatz von ADS ist allerdings die Verwendung der Serviceeinträge (*SRV-Records*), mit denen eine direkte Verbindung zu wichtigen Diensten innerhalb der ADS hergestellt wird. So werden über die SRV-Records beispielsweise Domaincontroller oder andere Diensteserver ermittelt (bei UNIX-Systemen funktioniert dies allerdings erst ab BIND-Version 8.1.2). Entsprechende RFCs (2136 und 2137) beschreiben die als DDNS (*Dynamic DNS*)

definierten dynamischen DNS-Aktualisierungen. Sowohl der Windows-Client als auch ein Windows-Server können beim DNS-Server die erforderlichen Registrierungen vornehmen. Eine enge Verzahnung mit dem DHCP-Server ist daher auch Bestandteil einer ADS- und DDNS-Implementierung.

➤ und Nachteile

In einer kurzen Übersicht sollen Vorteile und Nachteile des *Active Directory Service* einander gegenübergestellt werden:

- Vorteile
 - zentrale Verwaltungskonsole des gesamten Unternehmensnetzwerks (Einsatz von Gruppenrichtlinien)
 - einheitliche Anmeldung für alle Benutzer
 - einfache Integrierbarkeit bestehender Windows-Server mit älteren Betriebssystemen
 - Einsatz von Kerberos zur Absicherung notwendiger Vertrauensstellungen innerhalb der *Active Directory Services*
 - rasche Anpassungen an organisatorische Veränderungen innerhalb des Unternehmens durch flexible Strukturen
- Nachteile
 - *Active Directory Services* ist in Unternehmen mit Microsoft-unabhängigen Plattformen nicht einsetzbar – eine

Umstellung von UNIX-basierten Systemen ist ohne Windows-Server nicht möglich.

- Active Directory erfordert den Einsatz bestimmter DNS-Implementierungen (DDNS ist nicht unbedingt erforderlich, wird allerdings empfohlen), die zumindest Service-Ressourceneinträge (*SRV Records*) unterstützen müssen.
- Active Directory erfordert eine globale Anpassung sämtlicher Administrationsaktivitäten und damit eine unter Umständen geänderte Vorgehensweise bei der Wahrnehmung bisheriger Serviceleistungen gegenüber den Benutzern.
- Die Zusammenführung »alter« Windows-Domänen erfordert ein Höchstmaß an Sorgfalt und ausreichend geplante Mechanismen zur Realisierung eines möglichst störungsfreien Übergangs.

Auswahl der Betriebssystemplattform

In der Praxis wird heute in allen wichtigen Betriebssystemen das DNS als Dienst zur Namensauflösung verwendet. Selbst Microsoft empfiehlt bei Neu-Installationen den Einsatz von DNS und nicht mehr WINS, der viele Jahre den von Microsoft bevorzugten eigenen Dienst repräsentierte.

Mittlerweile erfolgt die Konfiguration von DNS auch über komfortable grafische Benutzeroberflächen, sodass einer raschen Einrichtung von DNS auf verschiedenen Betriebssystemplattformen nichts mehr im Wege steht.

Namensauflösung in der Praxis

Der Einsatz eines DNS-Systems bedingt das Einrichten bzw. die Verfügbarkeit eines Servers (DNS-Server) und die entsprechende Konfiguration der einzelnen Clients (DNS-Clients).

Vorgaben und Funktionsweise

Auf eine Detailbetrachtung der verschiedenen DNS-Servertypen (*primary, secondary, slave, forwarder, cache*) soll an dieser Stelle verzichtet werden; hier gilt der Hinweis auf weiterführende Literatur. Nachfolgend soll der Schwerpunkt auf der Einrichtung eines *Primary Name Server* (PNS) und eines *Secondary Name Server* (SNS), dem für einen Ausfall des PNS erforderlichen Backup-Systems, liegen.

Grundsätzlich erfolgt die Kommunikation innerhalb des DNS zwischen dem DNS-Server und dem DNS-Client (*Resolver*) in Form eines »Frage-und-Antwort-Spiels«. Kennt der lokale DNS-Server die IP-Adresse des angefragten Namens nicht, so

muss er sich bei anderen, ihm bekannten Nameservern danach »erkundigen«.

Der grundsätzliche Ablauf einer DNS-Anfrage ist [Abbildung 5-1](#) zu entnehmen. Wenn die *Root-Server* die Anfrage des lokalen DNS-Servers nicht beantworten können, so teilen sie dem lokalen DNS-Server die Zuständigkeiten (*Referrals*) für die angefragte Domain der ersten Hierarchiestufe mit (so z.B. den Nameserver des *Deutschen Network Information Centers*, wenn es sich um die Top Level Domain »de« handelt). Wenn auch dieser Nameserver den angefragten Namen nicht in eine ihm bekannte IP-Adresse umsetzen kann, so kennt er doch zumindest den Nameserver, der für die Second Level Domain zuständig ist. (Wenn beispielsweise der Host-Name [www.meier.de](#) angefragt wurde, so ermittelt dieser Top-Level-Nameserver nunmehr den zuständigen Second-Level-Nameserver für die Domain [meier.de](#).) Wird dieser Server im nächsten Schritt kontaktiert, so liefert er dem ursprünglich angefragten lokalen DNS-Server die gewünschte IP-Adresse, da diese ja in seiner lokalen DNS-Datenbasis geführt wird. Nun kann der lokale DNS-Server die Anfrage seines Clients beantworten.

Die DNS-Anfrage verwendet grundsätzlich eine hierarchisch orientierte Abfragesequenz, die beim *DNS-Root-Server*, der

»Wurzel« aller DNS-Server, beginnt. Die Namen dieser Root-Server werden meist in einer Datei »cache« oder »root« bei jedem UNIX-System bzw. bei Konfiguration einer DNS-Domain in einem Windows-Server automatisch angelegt und sind weltweit eindeutig. Nachfolgend ist der Inhalt einer solchen Datei beispielhaft dargestellt:

```
; This file holds the information on root name
; servers needed to initialize cache of InterNIC
; domain name servers
; (e.g. reference this file in the
; "cache . <file>"
; configuration file of BIND domain name server
;
; This file is made available by InterNIC
; registration services under anonymous FTP
```

```
;

;   file           /domain/named.root

;   on server      FTP.RS.INTERNIC.NET

;   -OR- under Gopher at  RS.INTERNIC.NET

;   under menu     InterNIC Reg.Serv. (NSI)

;   submenu        InterNIC Reg.Archives

;   file           named.root

;

;   last update:   Aug 22, 1997

;   related version of root zone: 1997082200

;

;

; formerly NS.INTERNIC.NET
```

```
;

.           3600000 IN NS   A.ROOT-SERVERS.NET

A.ROOT-SERVERS.NET.  3600000  A   198.41.0.4

;

; formerly NS1.ISI.EDU

;

.           3600000  NS   B.ROOT-SERVERS.NET.

B.ROOT-SERVERS.NET. 3600000  A   128.9.0.107

;

; formerly C.PSI.NET

;

.           3600000  NS   C.ROOT-SERVERS.NET.

C.ROOT-SERVERS.NET. 3600000  A   192.33.4.12
```

;

; formerly TERP.UMD.EDU

;

. 3600000 NS D.ROOT-SERVERS.NET.

D.ROOT-SERVERS.NET. 3600000 A 128.8.10.90

;

; formerly NS.NASA.GOV

;

. 3600000 NS E.ROOT-SERVERS.NET.

E.ROOT-SERVERS.NET. 3600000 A 192.203.230.1

;

; formerly NS.ISC.ORG

;

. 3600000 NS F.ROOT-SERVERS.NET.

F.ROOT-SERVERS.NET. 3600000 A 192.5.5.241

;

; formerly NS.NIC.DDN.MIL

;

. 3600000 NS G.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET. 3600000 A 192.112.36.4

;

; formerly AOS.ARL.ARMY.MIL

;

. 3600000 NS H.ROOT-SERVERS.NET.

H.ROOT-SERVERS.NET. 3600000 A 128.63.2.53

;

```
; formerly NIC.NORDU.NET
```

```
;
```

```
.          3600000    NS   I.ROOT-SERVERS.NET.
```

```
I.ROOT-SERVERS.NET. 3600000    A   192.36.148.17
```

```
;
```

```
; temporarily housed at NSI (InterNIC)
```

```
;
```

```
.          3600000    NS   J.ROOT-SERVERS.NET.
```

```
J.ROOT-SERVERS.NET. 3600000    A   198.41.0.10
```

```
;
```

```
; housed in LINX, operated by RIPE NCC
```

```
;
```

```
.          3600000    NS   K.ROOT-SERVERS.NET.
```

K.ROOT-SERVERS.NET. 3600000 A 193.0.14.129

;

; temporarily housed at ISI (IANA)

;

. 3600000 NS L.ROOT-SERVERS.NET.

L.ROOT-SERVERS.NET. 3600000 A 198.32.64.12

;

; housed in Japan, operated by WIDE

;

. 3600000 NS M.ROOT-SERVERS.NET.

M.ROOT-SERVERS.NET. 3600000 A 202.12.27.33

; End of File

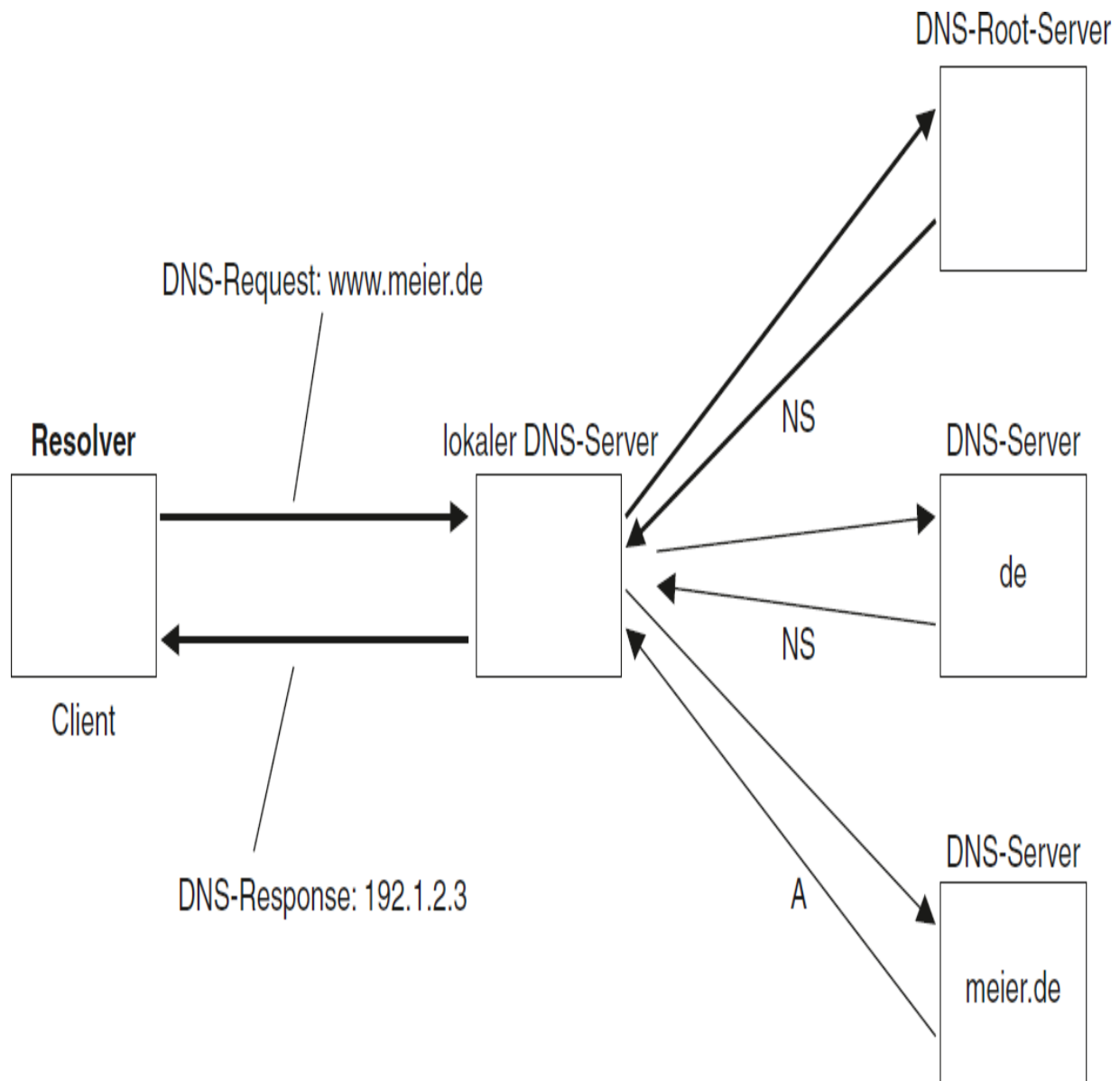


Abb. 5–1 Ablauf einer DNS-Abfrage

DNS-Konfiguration

Am Beispiel dient in einer UNIX-Betriebsumgebung die DNS-Serverinstallation als Implementierung des BIND (*Berkeley Internet Name Domain*). Während man unter den Windows-Implementierungen infolge der grafischen Benutzerführung

kaum mit den DNS-Ressourcendateien in Berührung kommt, ist es bei der Konfiguration unter UNIX-Systemen oft noch erforderlich, sich mit ihrer Definition und Bedeutung zu beschäftigen. Zahlreiche UNIX-Systeme besitzen allerdings heute auch umfangreiche und komfortable grafische Oberflächen, sodass die DNS-Konfiguration auch hier sehr erleichtert wird.

rcendateien

Nachfolgend werden die wesentlichen Funktionen der für die Installation und Konfiguration eines *Nameservers* relevanten Ressourcendateien erläutert.

- *Konfigurationsdatei* `/etc/named.conf`

Diese Datei enthält alle für den Serverbetrieb relevanten globalen Parameter und wird unmittelbar nach Start des *named Daemons* eingelesen. Der Inhalt einer solchen Datei ist nachfolgend beispielhaft dargestellt:

```
#  
  
# Globale Serverparameter
```

```
#

options {

directory "/var/named";

};

#

# Obligatorische Zonendatei für die Root-Serve

#

zone "." IN {

type hint;

file "named.root";

};

#
```

```
# Festlegen des Loopback (forward/reverse)

#

zone "localhost" IN {

type master;

file "named.localhost";

check-names fail;

allow-update { none; };

};

zone "0.0.127.in-addr.arpa" IN {

type master;

file "named.127.0.0";
```

```
check-names fail;
```

```
allow-update { none; };
```

```
};
```

```
#
```

```
# Die Masterdatei
```

```
#
```

```
zone "tcpip-grundlagen.de" IN {
```

```
type master;
```

```
file "named.tcpip-grundlagen.de";
```

```
allow-update { none; };
```

```
allow-query { any; };
```

```
notify yes;
```

```
};
```

```
#
```

```
# Festlegen der Reverse Resolution für beide S
```

```
#
```

```
zone "1.10.in-addr.arpa" IN {
```

```
type master;
```

```
file "named.10.1";
```

```
check-names fail;
```

```
allow-update { none; };
```

```
allow-query { any; };
```

```
notify yes;
```

```
};
```

```
zone "2.10.in-addr.arpa" IN {
```

```
type master;
```

```
file "named.10.2";
```

```
check-names fail;
```

```
allow-update { none; };
```

```
allow-query { any; };
```

```
notify yes;
```

```
};
```

- *Root-Server-Datei* `/var/named/named.cache`

In dieser Datei sind die derzeit aktuellen, weltweit eindeutig gültigen DNS-Root-Server definiert.

- *Forward-Lookup-Datei* `/var/named/named.localhost`

Das Mapping des *localhost* auf die Loopback-Adresse *127.0.0.1* ist in dieser Datei definiert, beispielsweise wie folgt:

```
@      1D IN SOA      @ root (
                                2000073001 ; serial
                                3H          ; refresh
                                15M         ; retry
                                1W          ; expiry
                                1D )        ; minimum

1D IN NS      @
```

1D IN A 127.0.0.1

- *Reverse-Lookup-Datei* `/var/named/named.127.0.0`

Hier erfolgt das Reverse-Mapping der Loopback-Adresse auf den Namen *localhost*:

```
$ORIGIN 0.0.127.in-addr.arpa.
```

@ 1D IN SOA localhost. root.localhost. (

2000073001 ; serial

3H ; refresh

15M ; retry

1W ; expiry

1D) ; minimum

1D IN NS localhost.

1 1D IN PTR localhost.

- *Forward-Lookup-Datei* `/var/named/named.tcpip-grundlagen.de`

In dieser Datei werden sämtliche in der Domäne *tcpip-grundlagen.de* definierten Rechnernamen den entsprechenden IP-Adressen zugeordnet:

@ IN SOA goofy root.localhost (

2000073001 ; serial

3H ; refresh

15M ; retry

1W ; expiry

1D) ; minimum

IN NS goofy

localhost IN A 127.0.0.1

goofy IN A 10.1.1.3

```
mickey          IN    A    10.1.1.2
```

```
firewall        IN    A    10.1.1.1
```

donald IN A 10.2.1.2

- *Reverse-Lookup-Datei* /var/named/named.10.1

Hier erfolgt das Reverse-Mapping der IP-Adressen des Subnetzes *10.1.0.0* auf die Rechnernamen:

@ IN SOA goofy.tcpip-grundlagen.de root.goofy (

2000073001 ; serial

3H ; refresh

15M ; retry

1W ; expiry

1D) ; minimum

IN NS goofy.tcpip-grundlagen.de.

1.1 IN PTR firewall.tcpip-grundlagen.de.

2.1 IN PTR mickey.tcpip-
grundlagen.de.

3.1 IN PTR goofy.tcpip-grundlagen.de.

- *Reverse-Lookup-Datei* /var/named/named.10.2

Hier erfolgt das Reverse-Mapping der IP-Adressen des Subnetzes *10.2.0.0* auf die Rechnernamen:

```
@      IN SOA      goofy.tcpip-grundlagen.de root.localhost (
```

```
2000073001      ; serial
```

```
3H      ; refresh
```

```
15M      ; retry
```

```
1W      ; expiry
```

```
1D )      ; minimum
```

```
IN NS goofy.tcpip-grundlagen.de.
```

```
1.1      IN PTR firewall.tcpip-grundlagen.de.
```

```
2.1          IN PTR donald.tcpip-  
grundlagen.de.
```

ce Records

Gemäß RFC 1035 vom November 1987 stehen für das Satzformat der *Resource Records* (RR) folgende Felder zur Verfügung:

- *NAME*

Name des IP-Knotens, zu dem dieser RR gehört

- *TYPE*

Ein zwei Bytes umfassendes Typenfeld mit folgenden Werten:

- A – IP-Adresse
- NS – autorisierter Nameserver
- MD – alte Angabe; neu MX
- MF – alte Angabe; neu MX
- CNAME – Angabe eines Alias-Namens
- SOA – gibt den Beginn einer Zone an (Start Of Authority)
- MB – experimentell
- MG – experimentell
- MR – experimentell
- NULL – experimentell

- WKS – Angabe von Diensten, den *well known services*
 - PTR – Angabe eines Zeigers auf eine Domäne
 - HINFO – Textinformation zur Beschreibung eines Hosts
 - MINFO – Mailbox- oder Mailing-List-Informationen
 - MX – Verteiler für Mail-Dienste
 - TXT – Zeichenketten
- *CLASS*
Ein zwei Bytes umfassendes Typenfeld mit folgenden Werten:
 - IN – Internet-Protokollfamilie TCP/IP
 - CS – alte Angabe; wird nicht mehr verwendet
 - CH – die CHAOS-Klasse; wird nicht verwendet
 - HS – Hesiod; wird nicht verwendet
- *TTL*
Ein vier Bytes umfassendes Integer-Feld. Es gibt den Zeitraum (in Sekunden) für beim Nameserver erneut angeforderte DNS-Daten an. Wird hier kein Wert angegeben, so wird der entsprechende Eintrag im SOA-RR verwendet. Bei der Wahl dieses Wertes sollte man sorgfältig vorgehen. Bei einem zu niedrigen Wert wird nämlich die Leistungsfähigkeit des Servers durch häufiges Anfordern reduziert; hingegen wird die Reaktionszeit des Servers negativ beeinflusst, wenn der TTL-Wert (*Time To Live*) zu hoch angesetzt ist.

- *RDLENGTH*

Ein zwei Bytes umfassendes Feld, das die Länge des folgenden RDATA-Felds angibt

- *RDATA*

In Abhängigkeit des TYPE- und CLASS-Formats werden hier Daten mit variabler Länge zur Beschreibung der Ressource angegeben.

;

Nachdem die Konfiguration des DNS-Servers unter Linux abgeschlossen ist, kann der Dienst mit dem Kommando */sbin/init.d/named start* gestartet bzw. mit */sbin/init.d/named stop* gestoppt werden. Will man einen laufenden *named-Daemon* dazu veranlassen, die geänderten Ressourcendateien erneut einzulesen, ohne den *named-Daemon* zu stoppen, so geschieht dies mit dem Kommando *kill -HUP <Prozess-Id>*.

HINWEIS

Das entsprechende Protokoll liegt im Verzeichnis */var/log/messages*.

Client-Konfiguration

Unabhängig davon, welche Nameserver-Variante eingesetzt bzw. unter welchem Betriebssystem ein Nameserver zur Verfügung gestellt wird, unterscheidet sich die Client-Konfiguration eigentlich nur durch das eingesetzte Client-Betriebssystem.

In der Praxis erfolgt die Einrichtung von DNS-Clients heute meist dynamisch. Das bedeutet, es sind keine manuellen Einträge mehr erforderlich, sondern dem Client wird über DHCP der relevante DNS-Server mitgeteilt, den er dann auch in seiner Netzwerkkonfiguration aufnimmt und speichert (siehe [Abb. 5-2](#) am Beispiel der Hardwareeigenschaften im Bereich Netzwerk und Internet in Windows 11).

WLAN Eigenschaften

IP-Zuweisung: Automatisch (DHCP)

Bearbeiten

DNS-Serverzuweisung: Automatisch (DHCP)

Bearbeiten

SSID: FRITZ!Box

Kopieren

Protokoll: Wi-Fi 5 (802.11ac)

Sicherheitstyp: WPA2-Personal

Hersteller: Realtek Semiconductor Corp.

Beschreibung: Realtek 8821CU Wireless LAN 802.11ac USB NIC

Treiberversion: 1030.38.712.2019

Netzfrequenzbereich: 2,4 GHz

Netzwerkanal: 6

Verbindungsgeschwindigkeit
(Empfang/Übertragung): 72/72 (Mbps)

IPv6-Adresse: 2a00:6020:11c1:4500:c9b2:3a91:7686:ac38

Verbindungslokale IPv6-Adresse: fe80::c9b2:3a91:7686:ac38%14

IPv6-DNS-Server: fd00::de15:c8ff:fe23:c3cb (unverschlüsselt)

IPv4-Adresse: 192.168.178.43

IPv4-DNS-Server: 192.168.178.1 (unverschlüsselt)

Physische Adresse (MAC): E0-E1-A9-31-B7-EA

Abb. 5–2 *DNS-Server wird dynamisch über DHCP übertragen.*

Protokolle und Dienste

Eine Produktfamilie wie TCP/IP stellt von Haus aus bereits eine Vielzahl an Zusatzoptionen, Protokollen, Programmen und Diensten zur Verfügung, die die Arbeit für den Anwender vereinfachen. Und nicht zuletzt die Tatsache, dass TCP/IP das Standardprotokoll schlechthin ist, sorgt für ein mittlerweile kaum noch überschaubares Angebot. Mit den Erläuterungen dieses Kapitels soll versucht werden, etwas Licht und Struktur in die Angebotspalette zu bringen, indem die wesentlichen Protokolle und Dienste einer TCP/IP-Umgebung dargestellt werden.

TELNET

Anfang der 80er Jahre des vorigen Jahrhunderts bestand ein Netzwerk aus einer Vielzahl »dummer« Terminals, die über eine entsprechende Verkabelung mit einem oder mehreren Großrechnern verbunden waren. Eine eigene Intelligenz besaßen diese Bildschirme nicht. Dennoch existierte ein Bedarf an Kommunikation, der über die dedizierte Kabelverbindung zu lediglich einem Rechner hinausging. Da man keine zusätzlichen Kabel installieren wollte, um eine weitere Rechnerverbindung zu etablieren, musste eine Softwarelösung für diese Problematik entwickelt werden. Auf diese Art und

Weise wurde ein Anwendungsprotokoll entwickelt, das den Zugriff auf weitere Rechner ermöglichte, ohne neue physische Verbindungen zwischen den einzelnen Clients und den betroffenen Rechnern (Server) zu installieren. Lediglich die Rechner selbst mussten untereinander verkabelt sein. Dies war die Geburtsstunde von *TELNET* (*Terminal Emulation*).

Nach Einreichung der ersten Entwürfe für *TELNET* beim IAB wurde es im Mai 1983 zum formalen Standard erhoben (RFC 854). Dabei wurde *TELNET* auch als *Standard-Remote-Login-Dienst* bezeichnet, der den Zugriff auf einen entfernten Rechner in einer Weise ermöglicht, die dem Benutzer den Eindruck vermittelt, er arbeite mit der Tastatur, dem Bildschirm und der gesamten Rechnerumgebung des Fremdrechners. Er benutzt quasi ein virtuelles Terminal in Form einer *Spiegelung* des Zielrechners. Das Bearbeiten von Dateien und der Start von Anwendungen sind dabei ebenso möglich wie die Benutzung entfernter Peripherie (z.B. Drucker). TELNET-Sitzungen lassen sich natürlich bidirektional etablieren, sodass zwei oder mehrere Anwender gleichzeitig in direkten Kontakt treten können.

HINWEIS

TELNET ist heute zwar immer noch fester Bestandteil der TCP/IP-Protokollimplementierungen verschiedener Rechnerarchitekturen, es wird aber mittlerweile zunehmend von der Secure Shell (ssh) ersetzt.

Telnet wurde letztmalig in den RFCs im November 2006 erwähnt. Dadurch, dass die Datenübertragung unverschlüsselt erfolgt, wird sein Einsatz nicht mehr empfohlen. Stattdessen wird die Secure Shell verwendet, deren Kommunikation (wie der Name bereits nahelegt) verschlüsselt wird. Dieser Dienst wird im folgenden Abschnitt näher erläutert.

SSH (Secure Shell)

Über SSH werden sichere Datenübertragungen sowie verschlüsselte Fernsitzungen realisiert. Wie im vorigen Abschnitt erwähnt, ersetzt SSH zunehmend die Telnet-Kommunikation.

Im RFC 4253 vom Januar 2006 wurde SSH erstmals vollständig beschrieben. In den folgenden Jahren (bis heute) erfolgten weitere Beiträge in entsprechenden RFCs. Die derzeit letzte Ergänzung erfolgte im RFC 9212 vom März 2022 (*Commercial National Security Algorithm Suite Cryptography for Secure Shell*). Die Kommunikation kann über Passworte oder

über Zertifikate erfolgen (letztere Option wird oft bei automatisch ablaufenden Skripten verwendet).

SSH-Server-Einrichtung

Da es sich bei SSH um eine Client-Server-Kommunikation handelt, muss zunächst auf der Server-Seite (in unserem später dargestellten Beispiel einer Fernsitzung ist dies der Rechner B) die SSH-Server-Software installiert und aktiviert werden.

Unter den Systemeinstellungen des Rechners B (Windows 10) werden bei den »Apps« die »optionalen Features« ausgewählt und in der Applikationsliste überprüft, ob ein SSH-Server bereits installiert ist. In unserem Beispiel muss dieser noch installiert werden.

Der OpenSSH-Server wird im Suchfeld eingegeben, gefunden und installiert. Anschließend ist er in der Applikationsliste aufgeführt (siehe [Abb. 6-1](#)).



Einstellungen



Optionale Features



Feature hinzufügen

Siehe Verlauf optionaler Features

Installierte Features

Installiertes optionales Feature suchen



Sortieren nach: Name ▾



Editor

316 KB



Internet Explorer 11

1,60 MB



Mathematik-Erkennung

16,6 MB



Microsoft Paint

3,34 MB



Microsoft-Remotehilfe

1,44 MB



OpenSSH-Client

5,05 MB



OpenSSH-Server

4,71 MB
26.04.2022

Abb. 6–1 OpenSSH-Serverinstallation

Damit der SSH-Server gestartet wird, muss sein Status in der Dienste-Liste überprüft werden. Dazu wird auf der Kommandozeile des Rechners B das Kommando *services.msc* eingegeben. Damit erscheint eine Übersicht aller Dienste und deren Status. Hier kann nun der OpenSSH-Server aktiviert werden (siehe [Abb. 6–2](#)). Nach seiner Aktivierung ist die Server-Einrichtung abgeschlossen.

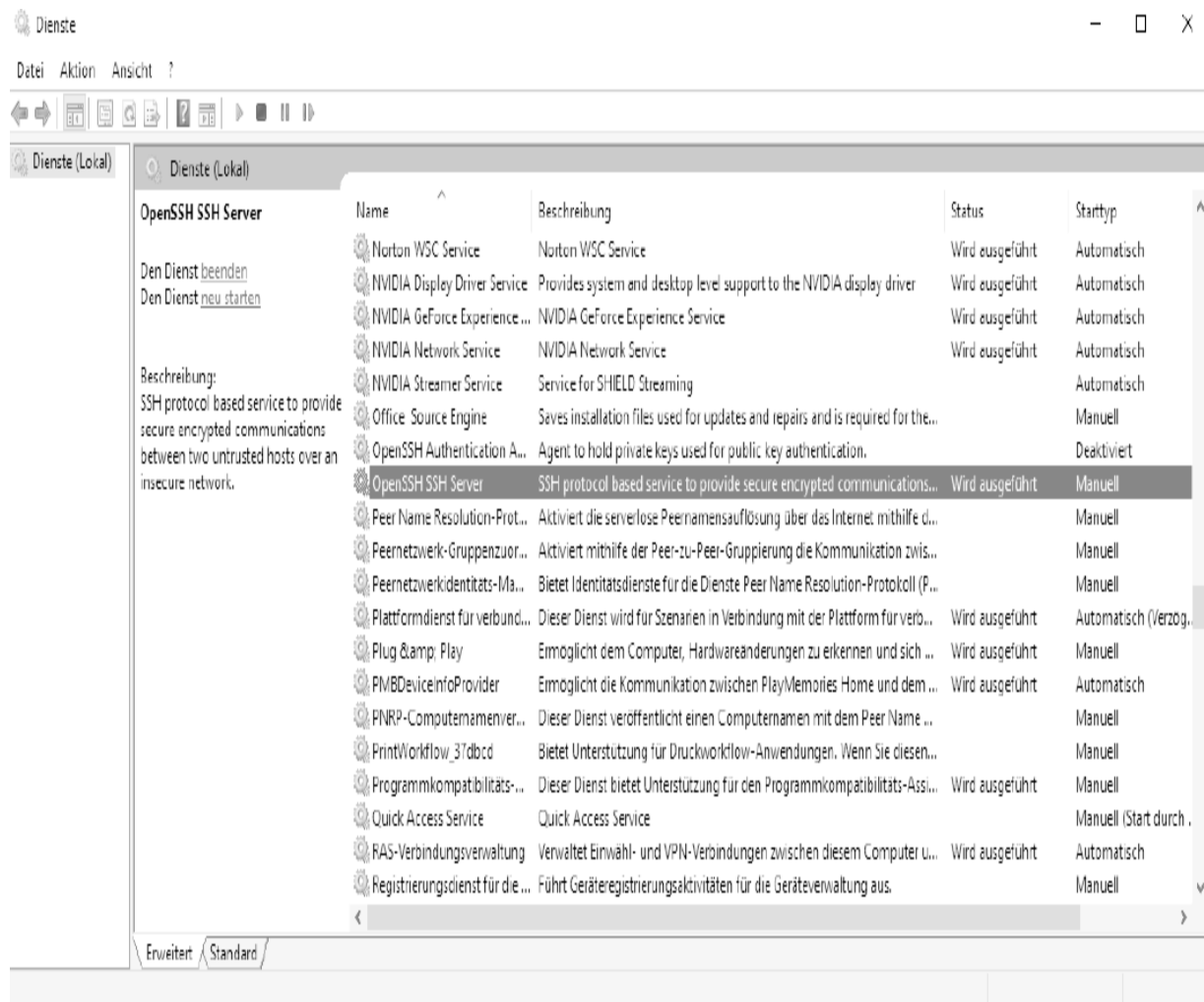


Abb. 6–2 Starten des OpenSSH-Serverdienstes

SSH-Client-Einrichtung

Auf der SSH-Client-Seite (Rechner A mit Windows 11) wird ein SSH-Client benötigt. In unserem Fall soll dies die im Internet frei verfügbare App »PuTTY« sein. Diese wird über ein grafisches Benutzer-Interface betrieben, sodass eine umständliche Bedienung im Kommandozeilen-Modus entfällt.

Nach Installation von PuTTY (das Programm wird in die Applikationsliste aufgenommen) wird das Programm gestartet und erwartet die Eingabe von Zielstation und Benutzernamen im Format *Benutzername@IP-Adresse* (siehe [Abb. 6-3](#)).

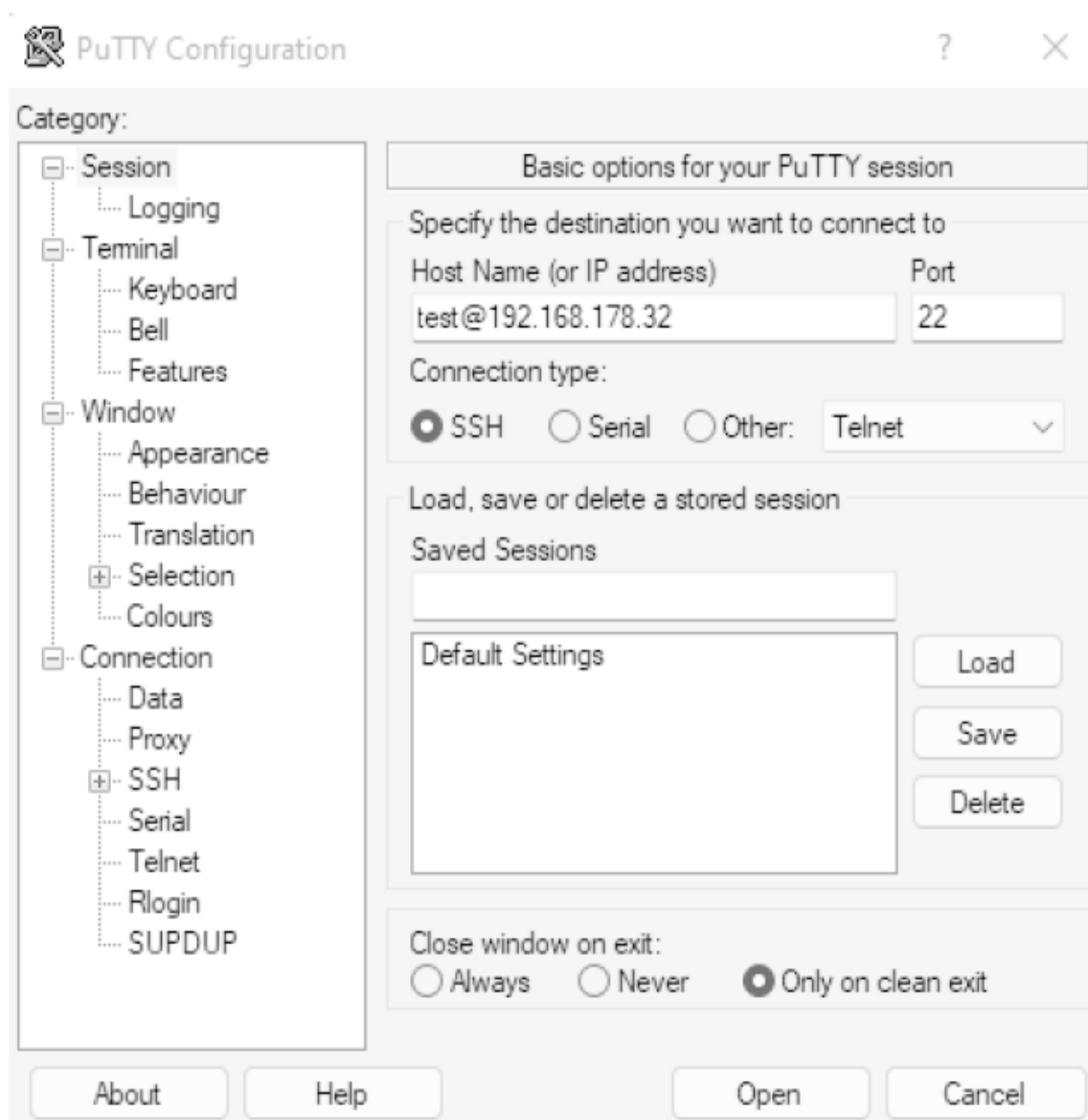


Abb. 6-3 PuTTY-Client verbindet sich mit *test@192.168.178.32*.

Der Benutzername muss einem gültigen Benutzerkonto auf dem Zielrechner B entsprechen. In diesem Fall wurde dazu der Benutzer »test« angelegt. In einem neuen Fenster wird nun das Passwort des Benutzers »test« eingegeben und es erfolgt der Zugriff auf das Hauptverzeichnis des Zielrechners B (siehe [Abb. 6-4](#)).

```
(c) Microsoft Corporation. Alle Rechte vorbehalten.

test@DESKTOP-GCMKUAN C:\Users\test>dir
Datenträger in Laufwerk C: ist Acer
Volumeseriennummer: CE7B-1A06

Verzeichnis von C:\Users\test

26.04.2022  13:16    <DIR>          .
26.04.2022  13:16    <DIR>          ..
27.11.2020  17:43    <DIR>          Desktop
26.04.2022  13:16    <DIR>          Documents
16.07.2016  13:47    <DIR>          Downloads
06.02.2017  14:39    <DIR>          Favorites
10.09.2018  23:41    <DIR>          Links
16.07.2016  13:47    <DIR>          Music
16.07.2016  13:47    <DIR>          Pictures
22.10.2017  20:44    <DIR>          Roaming
16.07.2016  13:47    <DIR>          Saved Games
16.07.2016  13:47    <DIR>          Videos
               0 Datei(en),               0 Bytes
              12 Verzeichnis(se), 49.586.536.448 Bytes frei

test@DESKTOP-GCMKUAN C:\Users\test>
```

Abb. 6-4 Anzeige des Hauptverzeichnisses im Zielrechner

Hier handelt es sich um ein sehr einfaches Beispiel zur SSH-Einrichtung und Aktivierung einer Fernsitzung. SSH lässt sich auch für eine sichere Dateiübertragung per SFTP (*Secure FTP*) einsetzen. Dies wird in einem der nächsten Abschnitte erläutert.

Dateiübertragung mit FTP

Die Übernahme von Daten aus Großrechner-Datenbanken oder aus anderen Quellsystemen auf direktem Wege ist effizienter als die Pflege eines mit zentnerschwerem Papier vollgestopften Aktenschanks. Auch wenn man sich bereits seit Jahren über einen anderen Weg zur Datenübertragung Gedanken gemacht hat, fehlte es in der Vergangenheit teilweise an den softwaretechnischen Möglichkeiten innerhalb der Infrastruktur der Datenverarbeitung. Bei Einsatz proprietärer Netzwerkprotokolle (die Bindung an einen einzigen Hersteller zur Betreuung von Hard- und Software ließ keine Alternativen zu) war es oft unmöglich, zwischen verschiedenen Rechnerarchitekturen einfache Datenübertragungen zu ermöglichen. Erst die Öffnung nahezu aller namhaften Unternehmen im Hinblick auf die Verwendung standardisierter Netzwerkprotokolle wie TCP/IP konnte eine Wende in dieser Problematik herbeiführen. Dies war auch die Geburtsstunde des *File Transfer Protocol* (FTP); dabei war der RFC 959 zwar

schon recht betagt (Oktober 1985), erreichte jedoch in den Folgejahren neue Attraktivität.

Funktion

Bei FTP handelt es sich um eine faszinierend einfache Softwarearchitektur, die jedoch ein Maximum an Leistungsfähigkeit zu minimalen Kosten erzielen kann. Warum dies so ist und für welche Einsatzzwecke sich FTP in der Praxis anbietet, soll nachfolgend erläutert werden.

Das Hauptproblem beim Versuch, Dateien von einem Rechner A zu einem Rechner B zu übertragen, sind oft unterschiedliche Dateiformate auf beiden Systemen. Für diese Inkompatibilität sind zumeist die verschiedenen Betriebssysteme mit ihren proprietären Dateisystemen verantwortlich.

Das Besondere an FTP ist, dass damit ein Dienst zur Verfügung gestellt wird, der es erlaubt, innerhalb sämtlicher Betriebssysteme über entsprechende TCP-Verbindungen Daten zu übertragen und diese in den jeweils verwendeten Dateiformaten abzuspeichern. Grundlage der Kommunikation ist hier ebenfalls das Client-Server-Modell.

Auch wenn FTP eine einfache Lösung für die Dateiübertragung darstellt, ist der generelle Ablauf einer FTP-Session etwas komplexer und bedarf einer genauen Betrachtung der Kommunikation in fünf einzelnen Phasen, beginnend mit der Darstellung der Phase 1 (Verbindungsaufbau) in [Abb. 6-5](#).

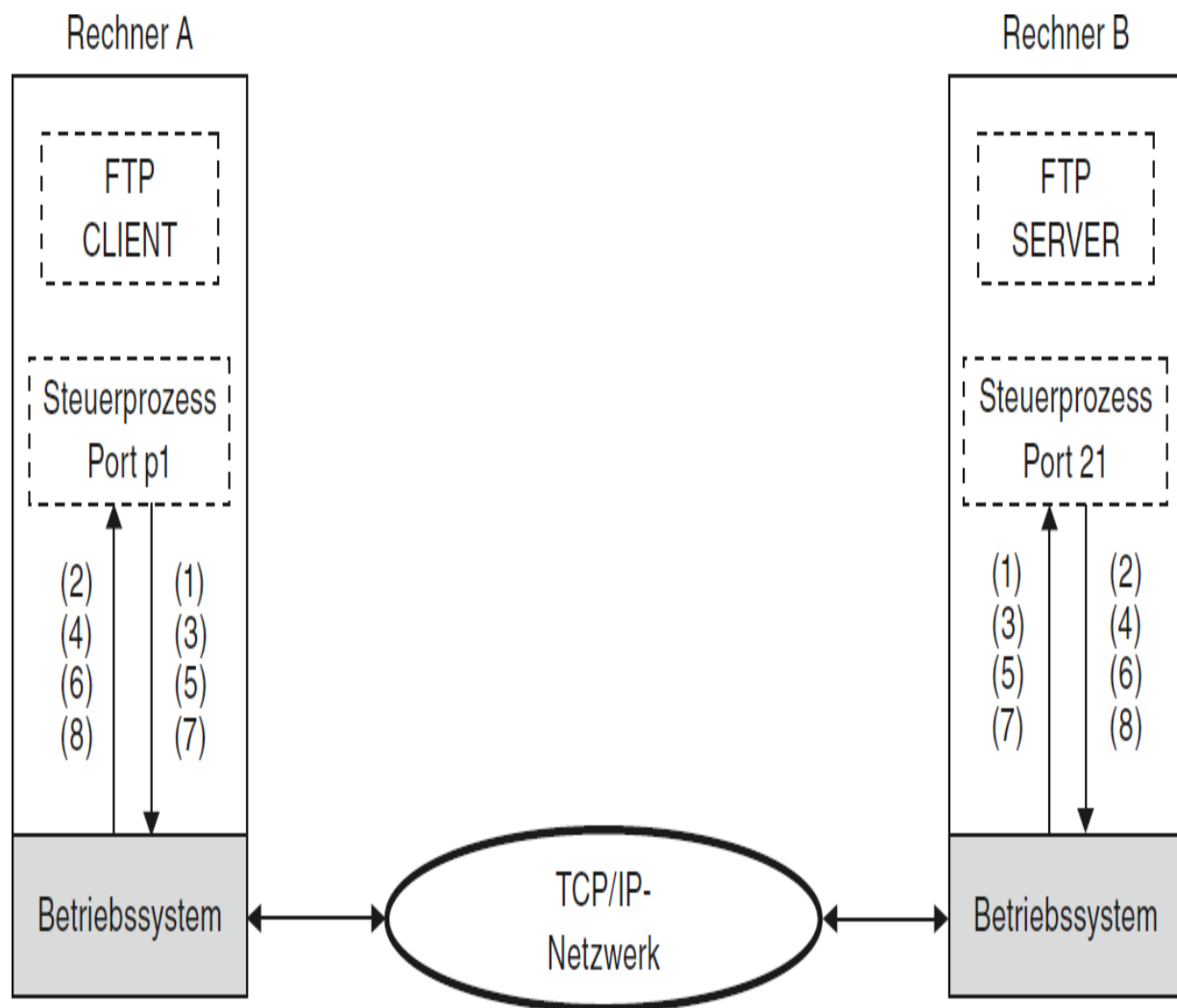


Abb. 6-5 Phase 1 – Verbindungsaufbau

Nach Starten der FTP-Client-Komponente auf Rechner A (FTP CLIENT) wird eine Steuerverbindung über den Serverport 21 aufgebaut (1). Die entsprechende FTP-Komponente auf Rechner B (FTP SERVER) beantwortet diese Aktion mit der Information, dass der gewünschte Dienst verfügbar ist (2). Nun sendet Rechner A über die Steuerverbindung den Benutzernamen (3). Rechner B empfängt diesen und bestätigt (4). Rechner A schickt das Kennwort (5). Rechner B empfängt und bestätigt, sofern die Kennworteingabe korrekt war (6). Die für die Datenübertragung möglichen Übertragungsoptionen (z.B. Binärformat) und der bzw. die Dateinamen werden von Rechner A übertragen (7). Rechner B empfängt und bestätigt (8).

Phase 2 des FTP-Prozesses beschäftigt sich mit der Herstellung der Datenverbindung und ist in Abb. 6-6 dargestellt.

In Phase 2 übermittelt der Rechner A an Rechner B über die Steuerverbindung die Anforderung einer Datenverbindung. Als Zielport wird Port *p2* angegeben (9). Daraufhin wird über den FTP-Serverprozess der Steuerverbindung auf Rechner B ein Serverprozess für die Datenverbindung mit Port-Nummer 20 erzeugt (10). Analog geschieht dies auch auf der Client-Seite. Hier wird der entsprechende Client-Prozess generiert (11). Nunmehr wird serverseitig von Rechner B mit der Ziel-

Portnummer $p2$ eine TCP-Datenverbindung für die eigentliche Datenübertragung aufgebaut.

Die eigentliche Datenübertragung erfolgt in Phase 3 des FTP-Prozesses und ist bildlich in [Abb. 6–7](#) dargestellt.

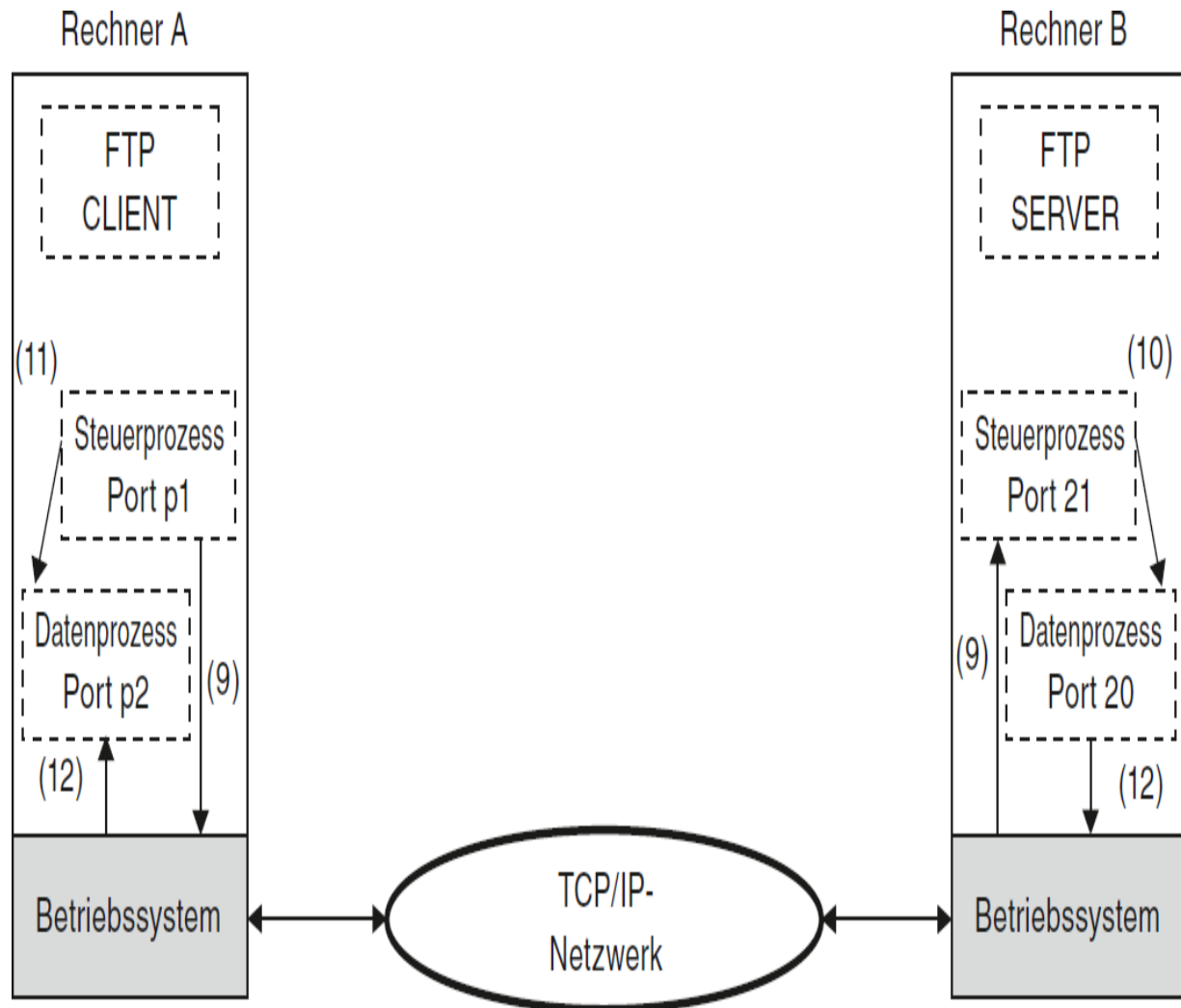


Abb. 6–6 Phase 2 – Generierung der Datenverbindung

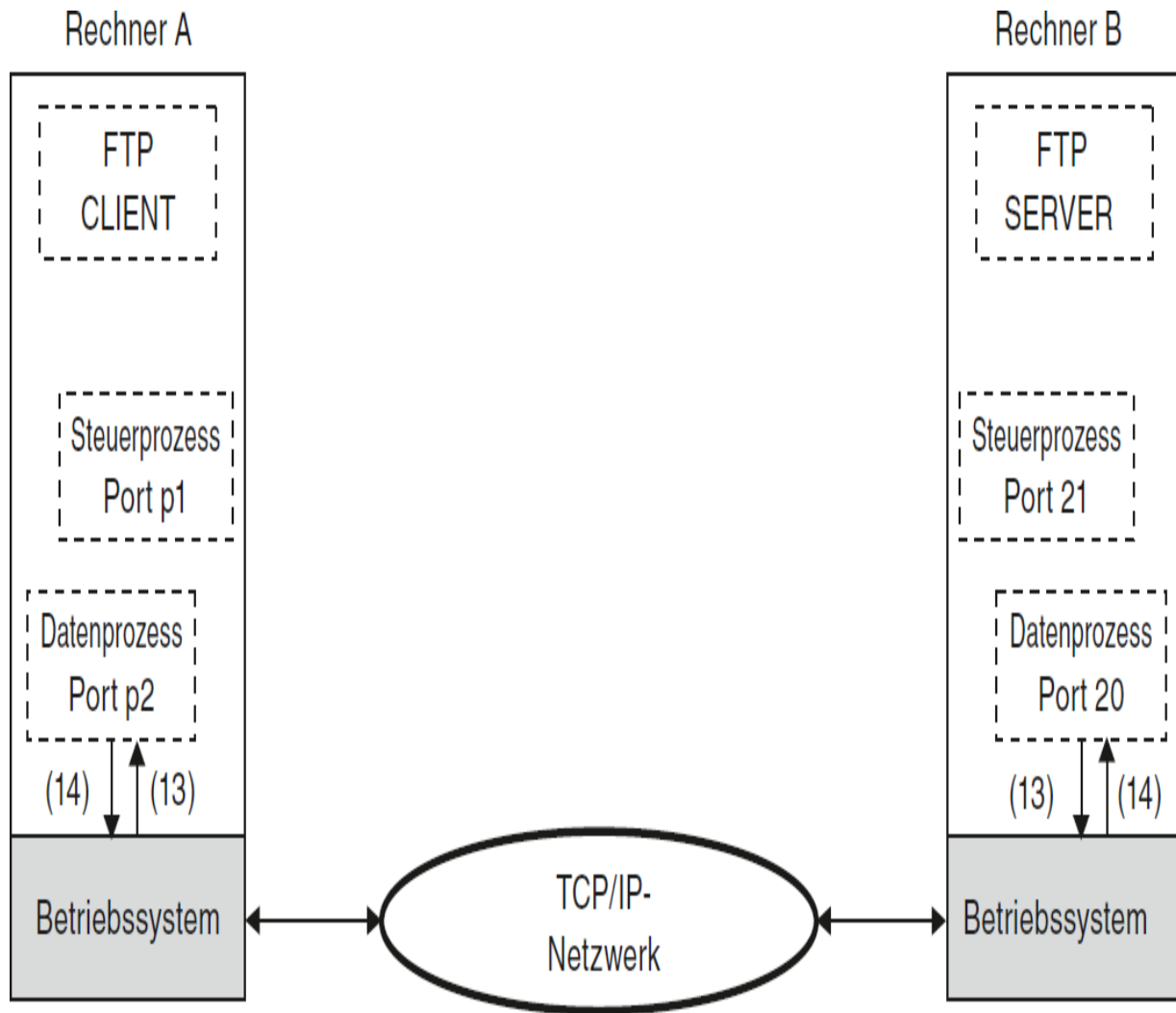


Abb. 6-7 Phase 3 – Datenübertragung

Der Server-Datenprozess liest bufferweise Datenblöcke aus der (vom Client) gewünschten Datei und überträgt diese über die TCP-Verbindung in gesicherten Segmenten zum Client-Datenprozess über das Netzwerk (13). Nach TCP-Protokollkonventionen werden die übertragenen Segmente per ACK (*Acknowledge*) bestätigt. Diese *Acknowledgements* erfolgen nach jedem einzelnen Segment (ohne *Windowing*-Verfahren)

oder nach einer bestimmten Anzahl von übertragenen Bytes (14).

Nach erfolgter Datenübertragung kann in Phase 4 das Ende der Übertragung definiert werden (siehe [Abb. 6–8](#)).

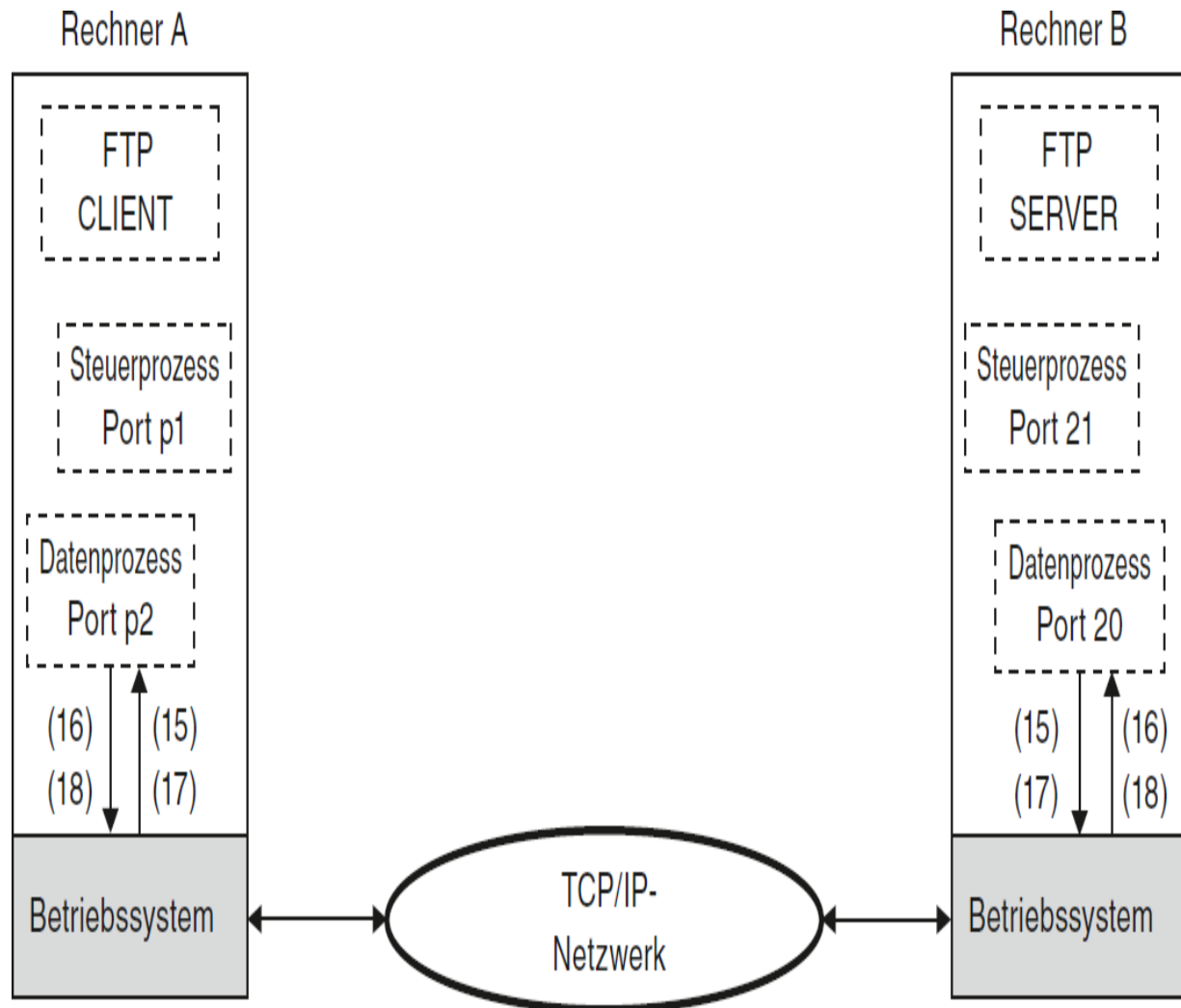


Abb. 6–8 Phase 4 – Einleitung des Übertragungsendes

Mit der Einleitung des Übertragungsendes in Phase 4 überträgt der Serverprozess die restlichen Daten (15). Der Client bestätigt

den korrekten Empfang der vollständigen Datei (16). Der Server-Datenprozess leitet nach erfolgreichem Dateitransfer den Abbau der Datenverbindung ein und teilt dies dem Client über einen *Close*-Befehl mit (17). Der Client-Datenprozess empfängt den *Close* und akzeptiert den Verbindungsabbau (18).

In der letzten Phase des FTP-Prozesses wird das Ende der Datenübertragung angekündigt, was in [Abb. 6–9](#) beispielhaft dargestellt ist.

Hier zeigt der Server-Datenprozess auf Rechner B seinem Kontrollprozess 21 das Ende der Datenverbindung an und wird beendet (19). Nachdem der Client-Datenprozess seinen Kontrollprozess informiert hat (20), wird er auch beendet.

HINWEIS

Auch nach dem Ende der Datenübertragung ist die Kontrollverbindung immer noch aktiv und könnte theoretisch anschließend weitere Datenübertragungen durchführen.

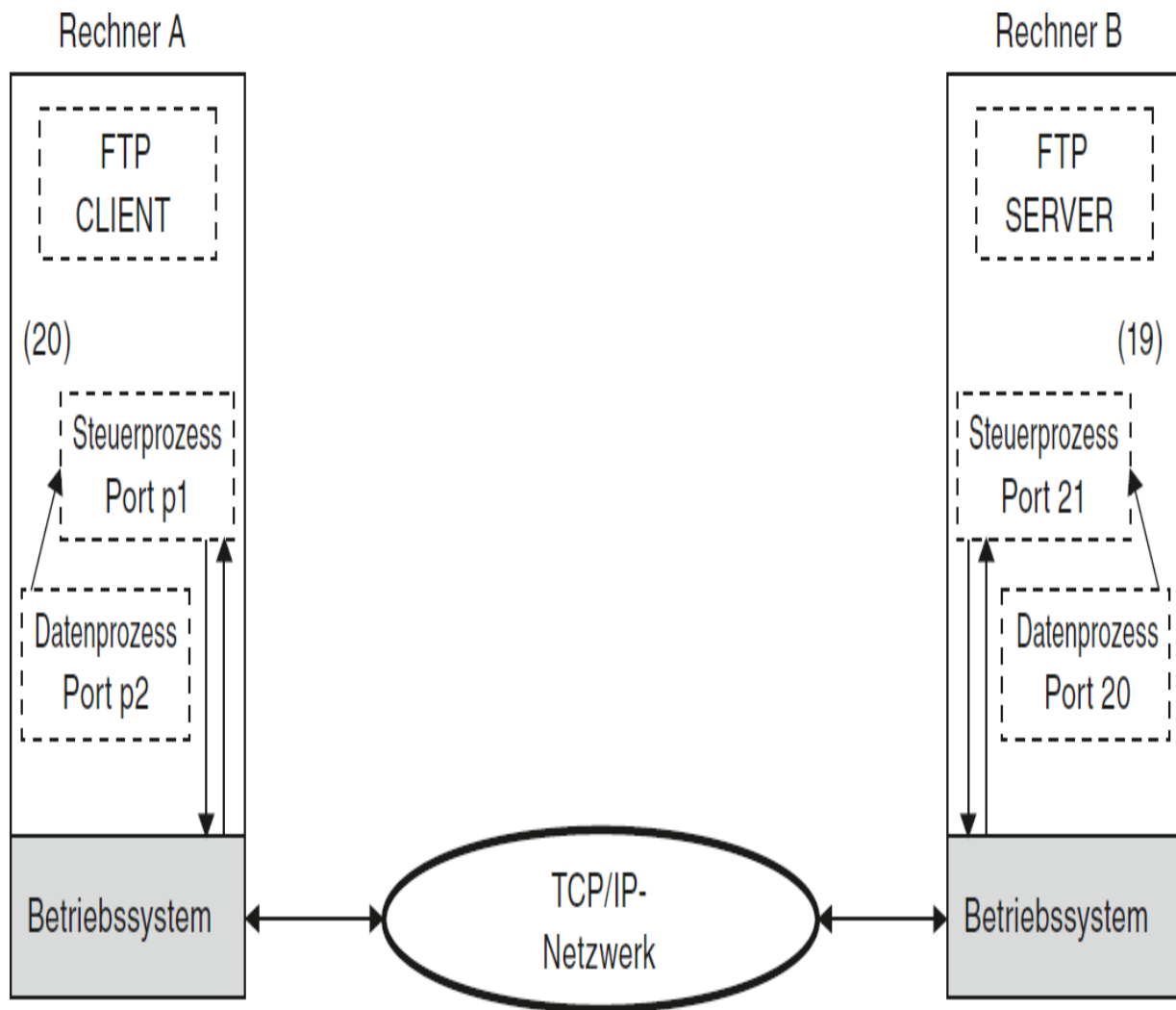


Abb. 6–9 Phase 5 – Ende der Datenübertragung

Ähnlich dem TELNET gibt es auch für FTP eine mittlerweile bevorzugte sichere Variante in der Datenübertragung: *Secure FTP* (SFTP); ebenfalls auf SSH basierend. Eine nähere Betrachtung erfolgt im nächsten Abschnitt einschließlich der Darstellung eines Beispiels.

Sicheres FTP (FTPS und SFTP)

Neben der zuvor behandelten unsicheren Ur-Version des FTP wurden in den letzten Jahren auch sichere Implementierungen entwickelt, die eine verschlüsselte FTP-Dateiübertragung gewährleisten.

Der RFC 4217 vom Oktober 2005 beschreibt eine sichere FTP-Variante, die auf TLS (*Transport Layer Security*) basiert, einer SSL-Implementierung von Netscape. Sie nutzt ebenso Sicherheitserweiterungen (zum Beispiel »starke Authentisierung«, Datenintegrität und –vertraulichkeit zwischen den FTP-Kommunikationspartnern) des FTP, die im RFC 2228 vom Oktober 1997 ausgeführt sind. Diese FTP-Implementierung wird auch FTPS (*FTP Secure*) genannt. Die Kommunikation erfolgt hier – wie bei FTP – über zwei Verbindungen.

Eine andere sichere FTP-Variante verwendet zur Absicherung der Kommunikation die Secure Shell (SSH). Bei diesem Verfahren wird das FTP-Protokoll durch eine einzige SSH-Sitzung »getunnelt«. Das folgende Beispiel basiert ebenfalls auf dem OpenSSH-Serverdienst und verwendet als Client-Software WinSCP. [Abb. 6–10](#) zeigt den Status des WinSCP-Clients

vor Verbindungsaufbau, [Abb. 6–11](#) zeigt das Ergebnis nach dem Verbindungsaufbau.

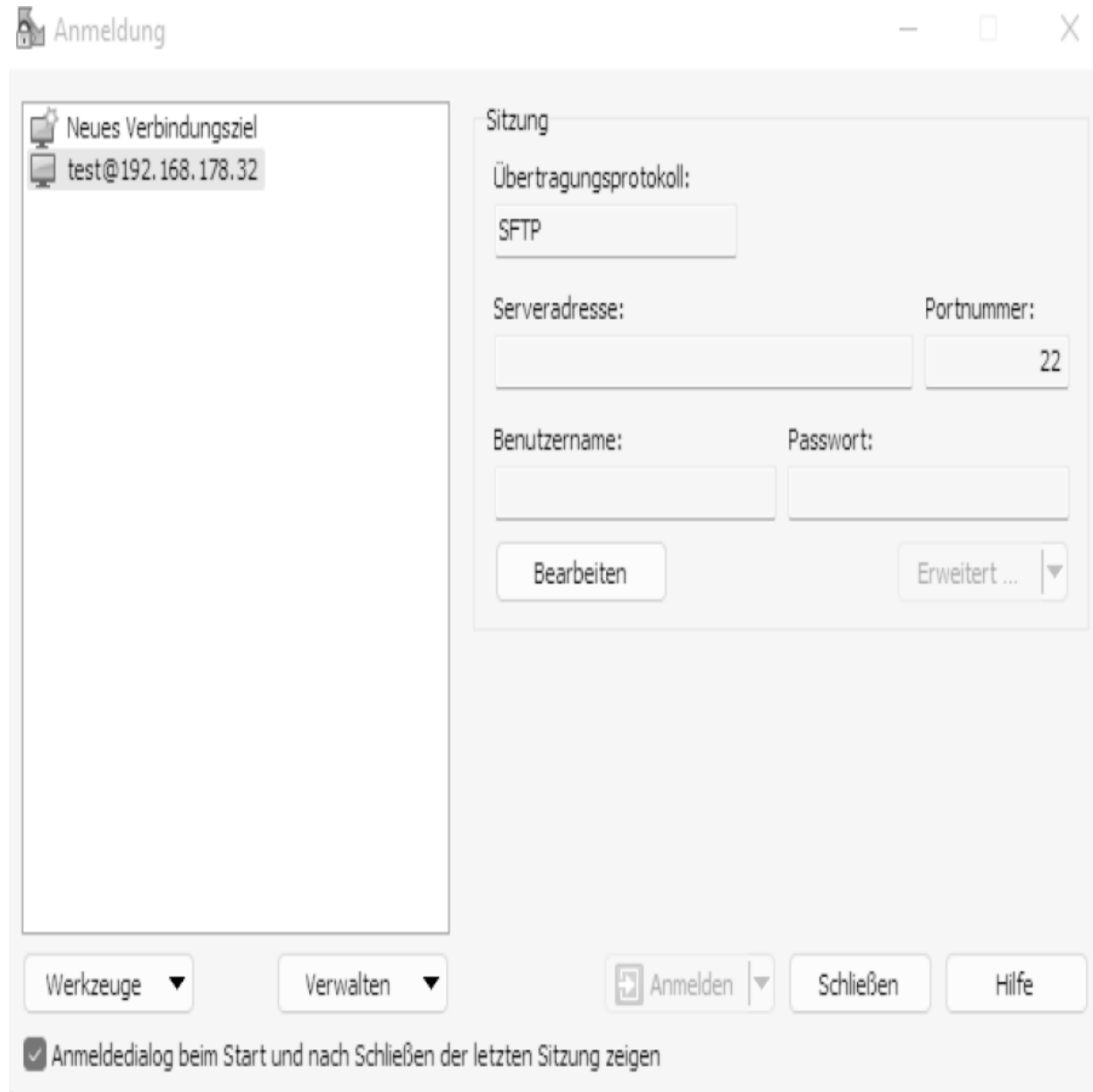


Abb. 6–10 WinSCP-Client VOR Verbindungsaufbau

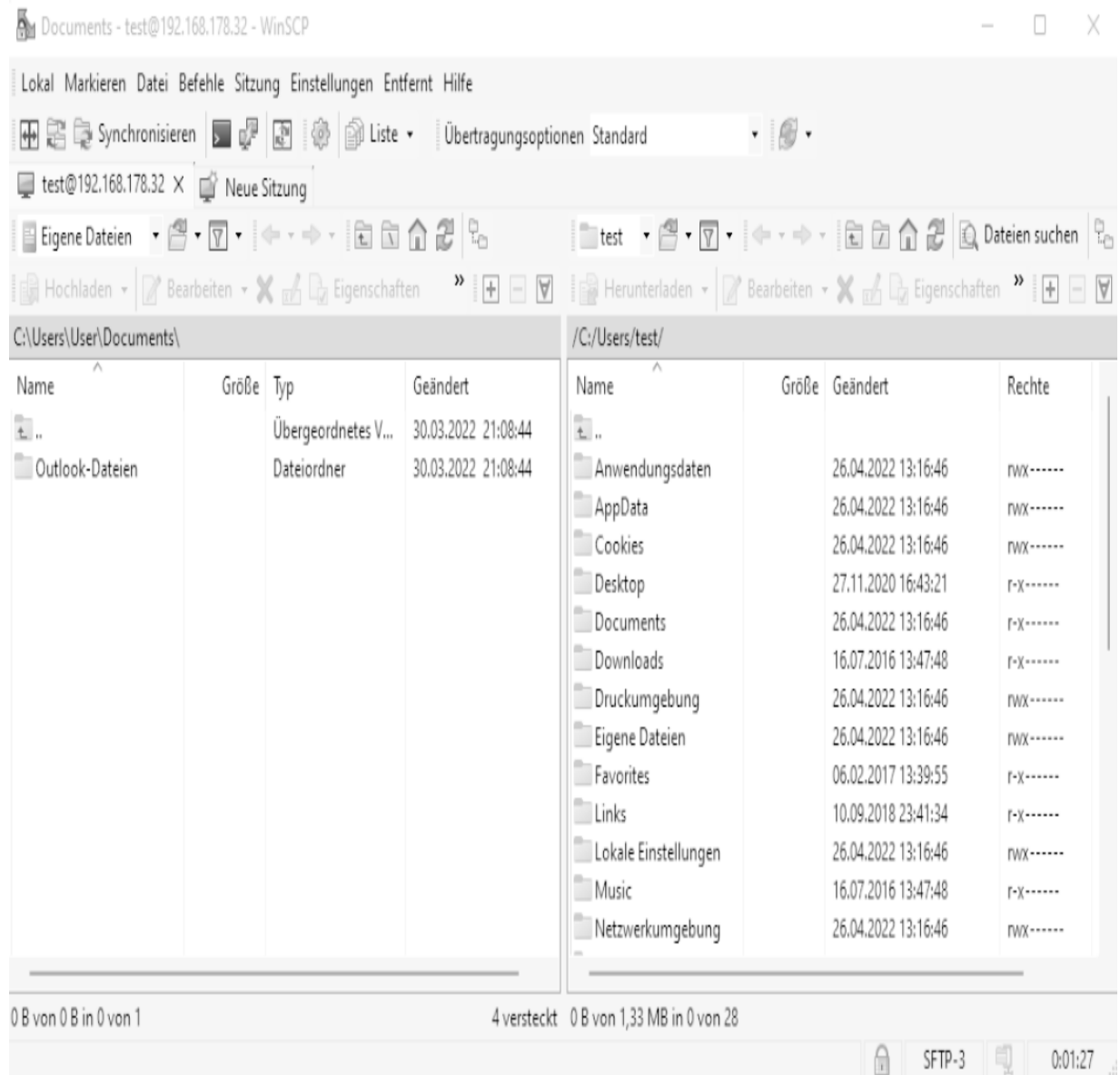


Abb. 6–11 WinSCP-Client NACH dem Verbindungsaufbau

Anonymus FTP

Bei der Arbeit bzw. Datenübertragung im Internet wird sehr oft die FTP-Konzeptvariante *Anonymous FTP* eingesetzt. Diese ermöglicht den nahezu völlig freien Zugang zu öffentlich betriebenen FTP-Servern.

Nach erfolgreicher Anmeldung am entfernten Host durch den User *anonymous* und durch Übergabe *der eigenen User-ID* als Kennwort ist die für einen Dateitransfer erforderliche FTP-Verbindung etabliert.

Trivial File Transfer Protocol (TFTP)

Im Gegensatz zum TCP-basierenden FTP stellt das TFTP ein Grundlagenprotokoll über UDP dar, das zwar ebenso wie FTP für die Dateiübertragung konzipiert wurde, allerdings nicht über seinen Sicherheitsstandard verfügt. Für Sicherungsmechanismen, die bei FTP im Protokoll TCP abgewickelt werden, muss das *Trivial File Transfer Protocol* selbst sorgen.

Der Einsatzbereich beschränkt sich somit in der Regel auf das Herunterladen von Systemprogrammen oder Netzwerksoftware. Oft ist diese Funktionalität auch schon auf integrierten Bausteinen der Systeme (z.B. Netzwerkkomponenten) bzw. auf Netzwerkadaptern implementiert. TFTP ist nicht für die Benutzung durch den Endanwender gedacht. Dies ist auch schon daran erkennbar, dass kein Zugriff auf irgendein Sicherheitskonzept hinsichtlich Authentifizierung erfolgt (Benutzerkennung, Kennwort).

Ähnlich dem komplexeren FTP arbeitet TFTP ebenfalls im Client-Server-Betrieb. Die Implementierung des Servers wird zumeist mit dem sogenannten *TFTP-Daemon* realisiert (UNIX, Linux: TFTPd, Windows: TFTPd.EXE). Dabei öffnet auf der Seite des Clients ein *Request to Write* die Dateiübertragung zum Server (Port 69). Die Anfrage (*Request*) wird vom Server bestätigt und anschließend werden die Daten in festen Satzlängen von 512 Bytes übertragen. Dabei wird jeder einzelne Datensatz nummeriert und muss vom Server bestätigt werden. Das Dateiende wird dadurch angenommen, dass ein Satz, dessen Satzlänge kleiner als 512 Bytes ist, den Empfänger erreicht.

In der nachfolgenden Aufstellung sind die verfügbaren Pakettypen einer TFTP-Verbindung aufgeführt:

Operation Code	Operation
1	Read Request (RRQ)
2	Write Request (WRQ)
3	Data (DATA)
4	Acknowledgement (ACK)
5	Error (ERROR)

In Ausnahmefällen werden in der Systemdatei *rhosts* IP-Adresse und eventuell die Benutzerkennung der zugelassenen

Netzteilnehmer hinterlegt, um über folgenden Befehlsumfang (die wichtigsten Befehle sind aufgeführt) zu verfügen:

- *connect*
Baut eine *Datagramm-Session* zum entfernten Host über das UDP-Protokoll auf.
- *mode*
Legt den Typ der Datenübertragung fest; zur Wahl stehen ASCII oder BINARY.
- *get*
Überträgt im Dialog (siehe auch FTP-Kommando) mit dem entfernten Host eine Datei von dort zum lokalen Rechner.
- *put*
Überträgt im Dialog mit dem entfernten Host eine Datei vom lokalen zum entfernten Host.
- *verbose*
Schaltet die Option ein bzw. aus, in der zusätzliche Informationen während der Dateiübertragung angezeigt werden.
- *quit*
Verlässt den TFTP-Dialogmodus.

HINWEIS

Treten während einer Übertragung Fehler auf, so wird ein Timer aktiviert. Läuft dieser ab, wird das zuletzt übertragene Paket wiederholt. Dabei ist es gleichgültig, ob es sich um ein Datenoder ein Acknowledgement-Paket handelt.

HTTP

Ein TCP-basiertes Protokoll, das insbesondere im Internet und Intranet zur Webkommunikation eingesetzt wird, ist das *HyperText Transfer Protocol* (HTTP). Während sich HTML (*HyperText Markup Language*) mit der Darstellung von Webseiten beschäftigt, übernimmt HTTP die Aufgabe, für einen reibungslosen Transport von HTML-Seiten zwischen Webserver und Web-Client zu sorgen.

HTTP ist als Protokollstandard für die Webkommunikation von allen Softwareherstellern akzeptiert und bildet die Grundlage des Datenaustauschs zwischen Webbrowser (auf dem Client) und einem Webserver. Zu diesem Thema gibt es mittlerweile eine Vielzahl von RFCs, die sich vorwiegend mit HTTP-Optimierungsmechanismen beschäftigen. Die Grundlage bildet heute der RFC 7235 (HTTP/1.1). Weiterentwicklungen zum HTTP/2 haben allerdings zu RFC 7540 vom Mai 2015 geführt bis zum derzeit letzten Update vom Februar 2020 (RFC 8740: *Using TLS 1.3 with HTTP/2*). Den aktuellen Stand bildet

heute jedoch HTTP/3; also HTTP in der Version 3. Diese Weiterentwicklungen werden in den letzten Abschnitten dieses Kapitels näher erläutert.

Eigenschaften

Während HTTP in der Version 1.0 noch verbindungslosen Charakter besaß, können ab der Protokollversion 1.1 mehrere *Request*- und *Response*-Aktivitäten parallel über ein und dieselbe Datenverbindung abgewickelt werden. Eine Verbindung braucht also nicht mehr abgebaut zu werden, wenn der Client dem Server eine Anfrage (*Request*) zur Bearbeitung gesendet hat. Dies hat wesentliche Vorteile:

- Durch die Beschränkung auf das Öffnen und Schließen einiger weniger TCP-Verbindungen werden CPU-Zeit sowie Speicherplatz für die Reservierung von TCP-Kontrollblöcken gespart.
- HTTP-*Requests* und -*Responses* können innerhalb einer Verarbeitung transportiert werden, sodass mehrere *Requests* übertragen werden können, ohne jeweils unmittelbar durch *Responses* beantwortet werden zu müssen.
- Die Netzwerkbelastung wird durch eine geringere Anzahl von TCP-Paketen (verursacht durch zahlreiche Verbindungsaufbausequenzen) deutlich reduziert.

- Statusmeldungen über eine Verbindung können weitergegeben werden und ermöglichen somit Reaktionen während einer aktiven HTTP-Kommunikation.

Adressierung

Da sich das World Wide Web aus einer sehr umfangreichen, weltweit auf Tausenden von Servern verteilten Ansammlung von Dokumenten und sonstigen Informationen zusammensetzt, bedarf es eines Adresskonzepts, um auf einzelne Dokumente zugreifen zu können. Ein entsprechendes Verfahren wird mittels *Uniform Resource Locator* (URL) zur Verfügung gestellt. Eine URL setzt sich dabei grundsätzlich immer aus folgenden Bestandteilen zusammen:

- Protokollangabe
- Portadresse des Kommunikationsprotokolls
- Pfadangabe
- Name des Dokuments

Die genaue Syntax sieht beispielsweise wie folgt aus:

<http://www.microsoft.com:80/windows/index.htm>

Dabei kann heutzutage die Angabe der Portadresse (:80) in der Regel entfallen, da der Standardport für die HTTP-Kommunikation 80 ist. So würde eine angepasste Syntax wie folgt lauten:

<http://www.microsoft.com/windows/index.htm>

Darüber hinaus ist es in einer URL auch möglich, direkt bestimmte Dokumente bzw. Dateien über weitere Protokolle anzusprechen, wie nachfolgend am Beispiel des FTP-Protokolls dargestellt:

<ftp://ftp.nic.de/pub/doc/rfc/rfc1900-1999/rfc1925.txt>

Auch in diesem Fall wird eine explizite Adressenangabe nicht benötigt, da die Portadressen 20 bzw. 21 (*well known*) für FTP auf allen öffentlichen Servern bereits vorgesehen sind.

HINWEIS

Eine URL, die in einem Browser verwendet werden kann, kann sich nicht nur auf das HTTP-Protokoll beziehen, sondern ermöglicht auch den Einsatz anderer Protokolle wie beispielsweise FTP.

Bei einer typischen HTTP-Kommunikation laufen im Einzelnen folgende Aktionen ab:

- Der Client richtet eine Anfrage (*Request*) an den Server, der einerseits den URL enthält, aus dem die vom Server angeforderte Ressource (z.B. eine HTML-Datei) hervorgeht, und andererseits weitere Informationen im HTTP-Header (z.B. Datum, Uhrzeit und Produktinformation des anfragenden Browsers).
- Der Server ist nun gehalten, festzustellen, ob die vom Client angeforderte Ressource überhaupt existiert und er diese zur Verfügung stellen kann. Anschließend wird von ihm der übertragene HTTP-Header verarbeitet.
- Der Server stellt die Datei-Extension fest und überprüft, ob es sich bei der angeforderten Ressource um einen gültigen MIME-Typ (*Multipurpose Internet Mail Extension*) handelt.
- Der Server überprüft den Seiteninhalt auf ASP-Skripte (ASP = *Active Server Pages*) und bringt diese zur Ausführung.
- Der Server bildet einen HTTP-Response-Header und übermittelt diesen an den Client. Dieser enthält einen

Statuscode, der den Status der Client-Anfrage beschreibt, sowie den MIME-Typ des zu sendenden Inhalts, damit der Client (bzw. der Webbrowser) weiß, wie er die empfangenen Daten behandeln muss.

HTTP-Message

Das HTTP-Protokoll stellt unterschiedliche Meldungstypen zur Verfügung, wobei grob zwischen folgenden *Message Types* unterschieden werden kann:

- Simple-Request (HTTP/0.9 Message)
- Simple-Response (HTTP/0.9 Message)
- Full-Request (HTTP/1.0 Message)
- Full-Response (HTTP/1.0 Message)

Die Typen *Full-Request* und *Full-Response* umfassen jeweils *Header Fields* und einen *Entity Body* gemäß RFC 822 (D. Crocker, »Standard for the format of ARPA Internet text messages«, 08/13/1982). *Header* und *Body* werden durch eine Leerzeile voneinander getrennt. Beim *Message Header* wird nochmals unterschieden zwischen *General Header*, *Request* (bzw. *Response*) *Header* und *Entity Header*.

Die HTTP-Meldungstypen *Simple-Request* und *Simple-Response* besitzen keinen *Header*. Sie werden deshalb infolge

des geringeren Informationsgehalts heute nur noch selten eingesetzt. Die Bedeutung der innerhalb von Syntaxregeln verwendeten Abkürzungen sind folgender Aufstellung (nach ANSI X3.4-1986) zu entnehmen:

OCTET = <any 8-bit sequence of data>

CHAR = <any US-ASCII character (octets 0 -

UPALPHA = <any US-ASCII uppercase letter "A"..

LOALPHA = <any US-ASCII lowercase letter "a"..

ALPHA = UPALPHA | LOALPHA

DIGIT = <any US-ASCII digit "0".."9">

CTL = <any US-ASCII control character (oct

CR = <US-ASCII CR, carriage return (13)>

LF = <US-ASCII LF, linefeed (10)>

SP = <US-ASCII SP, space (32)>

```
HT          = <US-ASCII HT, horizontal-tab (9)>
```

```
<">        = <US-ASCII double-quote mark (34)>
```

Die allgemeine Syntax zur Definition von *Message-Headern* ist nachfolgend dargestellt:

```
HTTP-header = field-name ":" [ field-value ] C
```

- *General Header*

Dieser Header wird sowohl bei einer *Request*- als auch bei einer *Response-Message* verwendet. Dazu gehören:

```
Date          = "Date" ":" HTTP-date
```

```
Pragma        = "Pragma" ":" 1#pragma-directive
```

- *Request Header*

```
Authorization = "Authorization" ":" credenti
```

```
From = "From" ":" mailbox
```

```
If-Modified-Since = "If-Modified-Since" ":"
```

```
Referer = "Referer" ":" ( absoluteURI | rela
```

```
User-Agent = "User-Agent" ":" 1*( product |
```

- *Response Header*

```
Location = "Location" ":" absoluteURI
```

```
Server = "Server" ":" 1*( product | comment
```

```
WWW-Authenticate = "WWW-Authenticate" ":" 1#
```

- *Entity Header*

```
Pragma = "Pragma" ":" 1#pragma-directive
```

```
Content-Encoding = "Content-Encoding" ":" co
```

```
Content-Length = "Content-Length" ":" 1*DIGI
```

```
Content-Type = "Content-Type" ":" media-type
```

```
Expires = "Expires" ":" HTTP-date
```

```
Last-Modified = "Last-Modified" ":" HTTP-dat
```

HTTP-Request

Eine Anfrage, die in Form einer *Request Message* vom Client zum Server geschickt wird, enthält in der ersten Zeile dieser *Message* die Angabe der Methode, die auf *Resource*, *Resource Identifier* und die Protokollversion angewendet werden soll. Zur Gewährleistung einer Rückwärtskompatibilität mit dem vom Funktionsumfang eingeschränkten HTTP/0.9 existieren zwei gültige Formate eines HTTP-Requests:

```
Request = Simple-Request | Full-Request
```

Simple-Request = "GET" SP Request-URI CRLF

Full-Request = Request-Line

*(General-Header

| Request-Header

| Entity-Header)

CRLF

[Entity-Body]

Wenn ein HTTP/1.0-Server einen *Simple-Request* erhält, muss dieser mit einer *HTTP/0.9-Simple Response* antworten. Ein HTTP/1.0-Client, der zum Empfang einer Full-Response in der Lage ist, darf niemals einen *Simple-Request* generieren. Dabei beginnt die *Request-Line* immer mit der Angabe der Methode,

gefolgt von der *Request-URI* (URI = *Universal Resource Identifier*) und der Protokollversion. Sie wird durch ein CRLF (*Carriage Return Line Feed*) abgeschlossen. Die einzelnen Elemente werden per »Blank« (Leerzeichen) voneinander getrennt, wobei CR oder LF nicht erlaubt sind. Die Syntax lautet wie folgt:

```
Request-Line = Method SP Request-URI SP HTTP-V
```

Der Unterschied zwischen einem *Simple-Request* und der *Request-Line* eines *Full-Request* besteht in der Angabe der HTTP-Version und der Verfügbarkeit von weiteren Methoden, die über die GET-Methode hinausgehen. So identifiziert die *Request-URI* die Ressource, auf die der Request angewendet werden soll. Die Syntax dafür lautet:

```
Request-URI = absoluteURI | abs_path
```

Die beiden Optionen der *Request-URI* sind vom Typ der Anfrage (*Request*) abhängig. Die Formulierung *absoluteURI* ist lediglich dann erlaubt, wenn der *Request* an einen sogenannten *Proxy*,

also einen Server-Stellvertreter gerichtet ist. Der Proxy wird zum Versand des *Request* veranlasst und liefert eine *Response* zurück. Sollte es sich bei dem *Request* um ein GET oder HEAD handeln und eine vorherige *Response* wurde zwischengespeichert (*Caching*), so kann der Proxy auch mit dieser *Cached Message* antworten. Der *Request* wird vom Proxy entweder zu einem weiteren Proxy geleitet oder direkt zum Zielservers, der in *absoluteURI* angegeben ist. Ein Beispiel für ein *Request-Line* ist nachfolgend dargestellt:

```
GET http://www.microsoft.com/pub/projects.html HTTP/1
```

HTTP-Response

Sobald eine *Request-Message* empfangen und entsprechend interpretiert wurde, antwortet der Server mit einer *Response Message*. Analog sieht der Aufbau der HTTP-Response aus:

```
Response = Simple-Response | Full-Response
```

```
Simple-Response = [ Entity-Body ]
```

```
Full-Response    = Status-Line

*( General-Header

| Response-Header

| Entity-Header )

CRLF

[ Entity-Body ]
```

Die erste Zeile einer *Full-Response* bildet die *Status-Line*. Sie besteht aus der Angabe der Protokollversion, einem numerischen Statuscode und einer entsprechenden Beschreibung. Jede dieser Angaben wird durch ein Leerzeichen getrennt. *Carriage Return* oder *Line Feed* ist nicht erlaubt, wobei die korrekte Syntax wie folgt lautet:


```
Status-Line = HTTP-Version SP Status-Code SP R
```

HINWEIS

Eine *Simple-Response* sollte nur dann generiert werden, wenn sie auf einen *HTTP/0.9-Simple-Request* antworten muss oder der Server nur HTTP/0.9 unterstützt.

Statuscodes

Im Zusammenhang mit dem HTTP-Protokoll wurde in den vorhergehenden Abschnitten mehrfach auf die Statusmeldungen (*Statuscodes*) hingewiesen. Dabei erfolgt grundsätzlich eine Einteilung dieser Statuscodes in die folgenden Kategorien:

- 1xx
Nur zur Information – ist für spätere Verwendung reserviert.
- 2xx
Erfolg – Die Aktivität wurde erfolgreich empfangen, verstanden und akzeptiert.
- 3xx

Umleitung – Es müssen weitere für eine Beendigung des Request erforderliche Aktivitäten durchgeführt werden.

- 4xx

Client-Fehler – Der Request beinhaltet Syntaxfehler und kann nicht bearbeitet werden.

- 5xx

Serverfehler – Der Server verweigert die Ausführung eines offensichtlich gültigen Request.

Methoden

In Zusammenhang mit http bezeichnen *Methoden* verschiedene Verfahren, nach denen Programmobjekte auf einem Webserver ausgeführt werden, und stellen darüber hinaus entsprechende Umgebungen zur Verfügung, die zur Parameterübergabe zwischen einem HTML-Dokument auf der Client-Seite und dem Webserver (wo sich das auszuführende Programm befindet) benötigt werden.

Für HTTP ist eine Reihe von Methoden definiert, die für die Realisierung der Verfahren zur Datenübertragung verwendet werden können. Am häufigsten kommen die GET- und die POST-Methode zum Einsatz. Die *GET-Methode* wird zumeist dort angewendet, wo eine geringe Datenmenge über Formulareingaben transportiert werden soll. Dabei wird das

URL-Encoding verwendet. Variablennamen und Werte werden in einer maximal 1024 Bytes umfassenden Zeichenkette durch das »&«-Zeichen voneinander getrennt. Leerzeichen werden entfernt und durch das »+«-Zeichen ersetzt. Die eigentliche URL wird vom Dateninhalt des Formulars durch das »?«-Zeichen getrennt. Ein Beispiel soll das verdeutlichen:

<http://www.testserver.de/form1.asp?var1=20&var2=30&vo>

HINWEIS

Bei der Übertragung von sensiblen Daten wie Benutzeridentifikationen oder Kennwörtern ist diese Methode unbedingt zu meiden, da die Eingaben für jedermann sichtbar der URL angehängt werden.

Die Codierung in einem HTML- oder ASP-Dokument erfolgt durch:

```
<form action="text.htm" method="GET">
```

```
<form action="text.asp" method="GET">
```

Die gesamten Formulareingaben befinden sich bei dieser Methode in der Umgebungsvariablen `QUERY_STRING`. Durch geeignete Algorithmen der Programmier- oder Skriptsprache müssen die Netto-Dateninhalte dadurch aus der Zeichenkette extrahiert werden, sodass Separatoren und Sonderzeichen (`>&<`, `>+<`, `>=<` usw.) entfernt werden. Das ASP-Konzept (*Active Server Pages*) stellt hierzu beispielsweise das `REQUEST`-Objekt *Request.QueryString* zur Verfügung. Das oben dargestellte Beispiel könnte demnach folgendermaßen entschlüsselt werden:

- `Request.QueryString ("var1")`
gibt den Wert *20* aus,
- `Request.QueryString ("var2")`
gibt den Wert *30* aus, und
- `Request.QueryString ("vorname")`
gibt den Wert *gerd* aus.

Kommt hingegen die *POST-Methode* zum Einsatz, werden Daten unter Verwendung des *HTTP-Headers* übertragen und an die Standardeingabe geschickt. Problematisch in diesem

Zusammenhang ist jedoch, dass eine Ende-Kennung des Datenstroms nicht zur Verfügung steht, sodass dieser nicht unmittelbar einer Variablen direkt zugeordnet werden kann. Hier muss die Umgebungsvariable `CONTENT_LENGTH` abgefragt werden, die den Datenumfang in Bytes enthält. Dies gilt in der Regel für CGI-Programme. Bei ASP-Dokumenten lassen sich mit dieser Methode die Dateninhalte des interaktiven Formulars direkt ansprechen.

HINWEIS

Weitere Methoden spielen im Weballtag keine große Rolle und werden an dieser Stelle nicht weiter erläutert. Detaillierte Beschreibungen sind dem RFC 2616 vom Juni 1999 zu entnehmen. Neuere Statuscodes sind in den letzten Jahren verwendet worden (z.B. 420 bis 426), haben aber bislang keinen Platz im RFC-Verzeichnis gefunden.

MIME-Datentypen

Die Übertragung von Daten zwischen einem Webbrowser und einem Webserver erfolgt unter Verwendung des MIME-Datentyps (*Multipurpose Internet Mail Extensions*). Dieser Standard stellt eine einheitliche Typisierung unterschiedlicher Datenformate im Internet sicher, ohne dass dedizierte

Umsetzungen oder Konvertierungen vorgenommen werden müssen. Diejenigen Datentypen, die diesem Standard nicht angehören, werden an Hilfsanwendungen, sogenannte *Plug-ins*, weitergeleitet und über diese entsprechend aktiviert. Auf diese Art und Weise entstanden innerhalb der Browsersoftware umfangreiche Sammlungen von Datentypen, die dem WWW den bekannten multimedialen Charakter verleihen.

Die nachfolgende Tabelle zeigt auszugsweise eine Übersicht definierter MIME-Datentypen. Die erste Veröffentlichung hierzu erfolgte im RFC 1341:

MIME-Datentyp	Dateierweiterung	Beschreibung
application/mac-binhex40	.hqx	Binhex-Datei
application/pdf	.pdf	PDF-Datei
application/postscript	.ps, .eps, .ai	PostScript-Datei
application/rtf	.rtf	Rich-Text-Datei
application/x-compress	.z	Komprimierte Datei
application/x-gzip	.gz	GZIP-Datei
application/x-hdf	.hdf	HDF-Datei
application/x-internet-signup	.ins, .isp	Internet Signup
application/x-stuffit	.sit	Stuffit-Archivdatei

application/x-tar	.tar	TAR-Archivdatei
application/x-x509-cert	.crt, .der	Site (CA) Certificate
application/zip	.zip	ZIP-Archivdatei
audio/basic	.au	Audiodatei
audio/x-aiff	.aiff, .aif	AIFF-Datei
audio/x-pn-realaudio	.ra, .ram	Real-Audiodatei
audio/x-wav	.wav	WAVE-Audiodatei
image/gif	.gif	GIF-Bilddatei
image/jpeg	.jpeg, .jpg	JPEG-Bilddatei
image/tiff	.tif, .tiff	TIFF-Bilddatei
image/x-xbitmap	.xbm	XBitmap-Bilddatei
text/html	.htm, .html	HTML-Dokument
text/plain	.txt, .c, .h, .ini	Textdatei
text/x-sgml	.sgml, .sgm	SGML-Dokument
video/mpeg	.mpeg, .mpg	MPEG-Videodatei
video/quicktime	.qt, .mov	QuickTime-Videodatei
video/x-msvideo	.avi	Microsoft-Videodatei

HTTP Version 2 (HTTP/2)

Trotz deutlicher Vorteile durch die Einführung der HTTP-Version 1.1 wurde unter dem Begriff HTTP/NG (*HTTP Next Generation*) bis 1998 weltweit an einer völligen Neukonzeption der HTTP-Kommunikation gearbeitet. Nunmehr haben sich seit der Einstellung dieser Arbeiten alle Aktivitäten zunächst auf die Entwicklung der HTTP-Version 2.0 konzentriert. Es sind im Internet zahlreiche Informationen zu diesem Thema zu finden. Ebenso die Planung einer HTTP-Standardisierung zur Version HTTP/3. Einzelheiten hierzu werden im nächsten Abschnitt dargestellt.

HTTP/3 und QUIC

Nachdem im Mai 2015 das HTTP-Protokoll in der Version 2 vom IETF als »proposed standard« veröffentlicht wurde, beschäftigt sich nun eine Arbeitsgruppe mit der Version 3. Einen RFC dazu gab es bei Drucklegung des Buches jedoch noch nicht.

In einer Übersicht sollen die Unterschiede der einzelnen Versionen dargestellt werden. Eine bedeutsame Besonderheit bildet hier das neue Protokoll QUIC für HTTP/3, das im RFC 9000 vom Mai 2021 erläutert wird. Es stellt für die HTTP-Kommunikation ein UDP-basiertes Transportprotokoll dar, das u.a. Mechanismen für parallele Datenströme (*stream*

multiplexing), Datenflusskontrolle für jeden Datenstrom (*per-stream flow control*) und einen schnellen Verbindungsaufbau (*low latency connection establishment*) zur Verfügung stellt. Der in älteren HTTP-Versionen übliche Datentransfer über TCP entfällt.

HTTP-Version	Charakteristika
HTTP/1 (1996)	▪ Verbindungslos ▪ Zustandslos ▪ Medienunabhängig
HTTP/1.1 (1997)	▪ Erster offizieller Standard ▪ Verbindung durch »keepalives« ▪ Pipelines (Bestätigung erst nach mehreren Anfragen) ▪ Caching
HTTP/2 (2015)	▪ Binärer Datentyp ersetzt Textdatei. ▪ Multiplexing = mehrere Anfragen parallel ▪ Header-Komprimierung = Vermeidung unnötiger Redundanzen

HTTP/3 (in der Entwicklung)

- UDP statt TCP basierte Übertragung = neues Protokoll QUIC

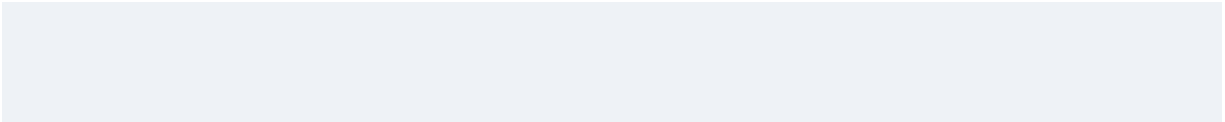
HTTPS

Unter HTTPS versteht man die sichere Variante des HTTP, bei der die Daten während ihrer Übertragung mit dem TLS-Protokoll (*Transport Layer Protocol*) verschlüsselt werden. In den Anfängen bis etwa 2014 erfolgte die Verschlüsselung mit dem SSL-Protokoll (*Secure Socket Layer*), wurde aber dann in der 2018 zum Standard erklärten TLS Version 1.3 abgelöst.

Die Verschlüsselung erfolgt über digitale Zertifikate, die von einer sog. »Certificate Authority« (CA) ausgestellt und über private und öffentliche Schlüssel dem Server und dem Client (meist einem Internet-Browser) für die Verschlüsselung zur Verfügung gestellt wird.

Die Syntax für die Adressierung einer Webseite in einer verschlüsselten http-Kommunikation sieht folgendermaßen aus:

<https://www.website.com/data-encrypt>



Ein Sicherheitsproblem bei dieser Kommunikation liegt darin begründet, dass über eine solche Kommunikation zwar die Identität der Kommunikationspartner garantiert ist, nicht aber deren Vertrauenswürdigkeit.

Für die Kommunikation wird bei HTTPS der Port 443 verwendet. Dies sollte man wissen, wenn man verschlüsselten http-Datenverkehr im Netzwerk identifizieren muss. Dies ist beispielsweise der Fall, wenn der Inhalt auf Schadcode bzw. Malware analysiert wird. Dabei erfolgt durch die Analysesoftware zunächst eine Entschlüsselung, anschließend die Analyse (und ggf. ein Entfernen der Malware) und zu guter Letzt eine erneute Verschlüsselung, bevor die Daten weitertransportiert werden. Dies ist zwar gängige Praxis, stellt aber hinsichtlich Datenintegrität im Sinne unserer Datenschutzgesetze (DSGVO) ein Problem dar. Auf der einen Seite fordert der Gesetzgeber, dass personenbezogene Daten von Unbefugten nicht ausgelesen werden dürfen, auf der anderen Seite sind seitens einer Unternehmung Maßnahmen zum Malware-Schutz unabdingbar. Die Lösung wird hier meist durch die gegenseitige Akzeptanz von Betriebsvereinbarungen

erreicht, in denen Arbeitgeber und Arbeitnehmer die Datenanalyse zur Bekämpfung von Malware vereinbaren.

E-Mail

Ein Dateitransfer bzw. der Austausch von Massendaten ist sicherlich nicht die übliche Kommunikationsform unter Menschen, sondern eher der Dialog, das Gespräch zwischen zwei Personen; dies entweder in persönlicher Form oder auch per Telefon. In der »computerisierten Welt« bedienen sich die Personen mittlerweile nicht selten eines speziellen Hilfsmittels, um mit einem Kommunikationspartner an einem anderen Rechner (überall auf der Welt) in Kontakt zu treten: *E-Mail* oder ausgeschrieben *Electronic Mail*. Diese Form der Kommunikation ist vergleichbar mit dem Abfassen eines Briefes, seinem Transport mit der Post und schließlich der Ankunft im Briefkasten des Empfängers. Mit E-Mail wird exakt dieses Konzept in die Welt der elektronischen Medien oder Möglichkeiten übertragen.

Kurze Zeit nach der Entstehung des Internets wurde der heute am meisten verwendete Dienst eingeführt: *E-Mail* (*Electronic Mail*). Mit E-Mail (fälschlicherweise auch oft als *eMail* geschrieben) können Nachrichten oder sonstige Daten

über das Internet an einen oder mehrere Empfänger versendet werden.

Beim E-Mail-Einsatz können eingehende Nachrichten archiviert, Adresslisten abgerufen und ausgehende Nachrichten an eine beliebige Zahl von Empfängern verschickt werden. Briefköpfe und Fußleisten sind speicher- und abrufbar, wobei die Nachricht auch jederzeit aus Textverarbeitungsprogrammen importiert werden kann.

Um eine E-Mail zu verfassen und über das Internet zu versenden, wird neben den softwaremäßigen Voraussetzungen lediglich grundlegendes Wissen zur Texteingabe vorausgesetzt. Darüber hinaus wird natürlich auch die E-Mail-Adresse des Empfängers benötigt, die wiederum weltweit eindeutig sein muss.

HINWEIS

Eine E-Mail-Adresse muss weltweit eindeutig sein und darf nicht doppelt vergeben werden. Sie setzt sich zusammen aus einem (eindeutigen) Namen und dem Domain-Namen (Organisation, Unternehmen oder auch Privatperson); getrennt durch das Zeichen @.

Ein wichtiger Grund für die explosionsartige Verbreitung von E-Mail ist, dass sie auf sehr einfachen, gut dokumentierten und seit vielen Jahren unveränderten Standards basiert. Der Kernstandard, auf dem das heutige Mail-System beruht (RFC 822), gilt noch immer und beschreibt E-Mails als reine Textdateien. Weltweit können Mail-Programme auf den unterschiedlichsten Rechnern und über Betriebssystemgrenzen hinweg miteinander kommunizieren. Die grafischen Benutzeroberflächen machen den Versand einer E-Mail extrem einfach.

Eine E-Mail besteht aus einem *Header* (Kopf), der Informationen über Absender und Empfänger enthält (entfernt vergleichbar mit einem Briefumschlag). Es folgt eine Leerzeile sowie der *Body* (Körper) mit der eigentlichen Nachricht. Die wichtigsten Header sind nachfolgend aufgeführt:

- *From:*
Absender
- *To:*
Empfänger
- *Date:*
Absendedatum
- *Cc:*

Weitere Empfänger, die die Mail als Durchschlag (*Carbon Copy*) erhalten

- *Bcc:*

Versand von heimlichen Kopien (*Blind Carbon Copy*), ohne dass der Empfänger dies mitbekommt. Die Header werden aus der Nachricht entfernt, sodass die anderen Empfänger nichts von der zusätzlichen Kopie mitbekommen.

- *Subject:*

Betreff der E-Mail

Der Körper (*Body*) einer E-Mail besteht aus beliebig vielen ASCII-Zeichen (7-Bit-Code); nur Zeichen mit Codes zwischen 0 und 127 sind zulässig. Ausführbare Programmdateien, Bilder oder Audiodateien können jedoch beliebige Bytes zwischen 0 und 255 enthalten. Um solch eine Binärdatei per E-Mail übertragen zu können, muss der Datenstrom so codiert werden, dass er sich im ASCII-Code abbilden lässt. Dies erfolgte in den Anfängen von E-Mail mit Programmen wie *uuencode*. Es wandelte die Dateien für den Mail-Versand in eine Folge druckbarer ASCII-Zeichen um. Das entsprechende Gegenstück *uudecode* stellte daraus wieder die Originaldateien her.

Die heutzutage eingesetzten aktuellen Mail-Programme haben mit Umlauten kein Problem mehr und können beliebige Dateien als Zusatz (*Attachments*) an E-Mails anhängen. Seit 1996

gibt es hierfür einen Standard namens *MIME* (*Multipurpose Internet Mail Extensions*). Er erweitert den RFC 822 und führt innerhalb des Bodys eine Struktur ein, die durch speziell formatierte Zeilen nach ähnlichem Muster wie im Header vorgegeben ist. Eine E-Mail lässt sich auf diese Weise in mehrere Teile aufgliedern, wobei jedem Teil selbst eine Art Header vorangestellt wird. Dieser gibt wiederum an, um welche Art von Information es sich handelt und wie sie codiert ist.

Auch wenn E-Mail heutzutage eine Standardanwendung darstellt und für den Anwender sehr einfach und leicht zu nutzen ist, sind damit dennoch einige technische Herausforderungen verbunden, die im Einzelnen nachfolgend erläutert werden. Der Schwerpunkt liegt dabei auf den Diensten und Protokollen, die die TCP/IP-Protokollfamilie zur Verfügung stellt.

Simple Mail Transfer Protocol (SMTP)

Neben den Inhalten bzw. Bestandteilen einer E-Mail ist ebenfalls von großem Interesse, wie die E-Mail vom Sender zum Empfänger gelangt. Basis dafür bildet das Protokoll mit dem Namen *Simple Mail Transfer Protocol* (SMTP), ein Protokoll der TCP/IP-Protokollfamilie. SMTP beschreibt einen einfachen Dialog, mit dem sich eine Mail über eine Punkt-zu-Punkt-

Verbindung zwischen zwei Systemen übermitteln lässt. Bezogen auf das Internet handelt es sich dabei um eine TCP-Verbindung zum SMTP-Port 25 des Zielrechners. Dabei weiß der Absender anhand des in der Mail-Adresse angegebenen Domain-Namens, wohin die E-Mail zu senden ist.

Historisch für den Bereich UNIX-basierter Systeme entwickelt, aber mittlerweile auch auf Computern mit den dort etablierten Betriebssystemen (Windows, Macintosh, Linux) verfügbar, hat sich das *Simple Mail Transfer Protocol* (SMTP) als Standardprotokoll für die E-Mail-Übertragung herausgebildet. Dabei beruht SMTP, wie viele andere Punkt-zu-Punkt-Protokolle auch, auf dem Client-Server-Konzept.

Bei Nutzung der E-Mail-Möglichkeiten bedient ein Anwender eine spezielle Software (*E-Mail-Client*) und bereitet die Nachricht, die er elektronisch versenden möchte, vor. Diese übergibt er anschließend (über den Mail-Client) an einen Mail-Prozess (Sender), der dafür sorgt, dass die Nachricht zwischengespeichert werden kann (*Spooling*). Über das TCP-Protokoll wird eine bidirektionale Verbindung zum Empfangsprozess auf dem entfernten Host (*Mail-Server*) aufgebaut (*well known Port 25*). Dabei bewahrt der Spool-Bereich die Daten so lange auf, bis sie erfolgreich an den Mail-Server übergeben worden sind.

Der Aufbau einer Mail-Adresse ergibt sich zumeist aus der Kombination von Domäne und Benutzeridentifikation. In einem Beispiel heißt der Benutzer *Gerd* und die Domäne *tcpip-netz.muenster.de*. Die vollständige Mail-Adresse hieße demnach: *Gerd@tcpip-netz.muenster.de*.

Folgende Befehle oder Anweisungen stehen auf der Client-Seite (E-Mail-Client) zur Verfügung:

HELO	Identifikation des Senders
MAIL	Start des Mail-Prozesses
RCPT	Identifikation des Empfängers
USER	Benutzer-ID wird bestätigt.
DATA	Datentransfer, Transferende mit <crlf>.<crlf>
RESET	Abbruch der Verbindung
NOOP	Bestätigung wird erwartet, keine Aktion.
QUIT	Verbindung wird ordentlich abgebaut.

Die Aktionen auf der Seite eines E-Mail-Servers sind auszugsweise nachfolgend dargestellt:

21	Service wird beendet.
50	OK
51	Benutzer nicht lokal, wird weitergeleitet an <xyz>
54	Start der Mail-Eingabe, Ende mit <crLf>.<crLf>
00	Syntaxfehler
50	Operation nicht möglich, Mailbox nicht erreichbar

Den exakten Ablauf einer Mail-Sitzung zeigt folgende Beschreibung:

- Der Client (bzw. Sender) baut eine Session zum Server (bzw. Empfänger) auf.
- Der Server schickt eine »Service Ready«- (Code 220) oder eine »Service Not Available«-Meldung (Code 421).
- Der Sender identifiziert sich mit *HELO* bzw. *EHLO*, der Empfänger antwortet mit seinem Domänen-Namen.
- Nun beginnt der Client mit dem eigentlichen Mail-Prozess, indem er den MAIL-Befehl absetzt. Der Server antwortet mit *OK* (Code 250).
- Der RCPT-Befehl übermittelt dem Server den bzw. die Empfänger, worauf der Server mit *OK* (Code 250) oder Code

550 – »mailbox unavailable« – antwortet.

- Der Client beginnt nun mit der Datenübertragung per DATA-Befehl. Der Server reagiert mit einem Start-Mail-Input, Ende mit `<crlf>.<crlf>` (Code 354).
- Soll die Übertragung beendet werden, so zeigt dies der Sender vereinbarungsgemäß mit der Zeichenfolge `<crlf>`. `<crlf>` an.
- Der Sender beendet die Verbindung mit dem QUIT-Befehl. Der Server antwortet mit »Service Closing« (Code 221).

Das SMTP- oder Mail-Protokoll für den zuvor beschriebenen Ablauf sieht folgendermaßen aus:

Client	Server
Verbindungsaufbau	
	 220 <tcpip-netz.ulm.de> Service Ready
HELO <tcpip-netz.muenster.de>	
	 250 <tcpip-netz.ulm.de> OK
MAIL FROM: <Gerd@tcpip-netz.muenster.de>	
	 250 OK

RCPT TO: <Karl@tcp-
netz.ulm.de>

⇒

⇐ 250 OK

DATA

⇒

⇐ 354 Start Mail Input, end
with <crlf>.<crlf>

Zeile 1

⇒

Zeile 2

⇒

Zeile 3

⇒

...

Zeile n

⇒

<crlf>.<crlf>

⇒

⇐ 250 OK

QUIT

⇒

⇐ 221 <tcpip-netz.ulm.de>
Verbindungsabbau

HINWEIS

Eine Mail-Sitzung kann derart variiert werden, dass nach erfolgter Übertragung der E-Mail der Sender und der Empfänger wechseln, bevor die Verbindung geschlossen wird. Mit der Anweisung TURN kann dieses Vorhaben vom Sender angezeigt werden.

SMTP wurde in den 70er und 80er Jahren des vorigen Jahrhunderts hauptsächlich in der UNIX-Welt eingesetzt. Durch das enorme Wachstum der weltweiten E-Mail-Aktivitäten im Internet ist SMTP auch sehr schnell in den Desktop-Bereich (Windows, Linux, macOS) und in den Bereich der Tablets und Smartphones eingezogen.

HINWEIS

Für das ISO/OSI-Umfeld gibt es ein funktionales Pendant: das *X.400-Message Handling System*. Dabei erfolgt über sogenannte *User Agents* (UA) der Transport von Nachrichten an die *Message Transfer Agents* (MTA), die wiederum in einem dichten Netzwerk untereinander verbunden sind. Das Leistungsspektrum dieses Systems ist um ein Vielfaches komplexer als beim SMTP; auf Details hierzu wird in diesem Buch nicht näher eingegangen.

Eine besonders leistungsfähige Erweiterung des SMTP hat für eine Aufwertung auch im Vergleich zu anderen Systemen gesorgt. Nach RFC 822 vom August 1982 konnten bis dahin lediglich textuelle Informationen als Mail innerhalb des SMTP verschickt werden. Als andere Mail-Systeme (z.B. X.400) in den 80er Jahren auch Daten binären Charakters in Umlauf bringen konnten, löste dies eine Kompatibilitätsproblematik aus, denn bestimmte Mail-Übergänge oder Mail-Weiterleitungen waren ohne Weiteres nicht zu realisieren. Umständliche Konverter mussten eingesetzt werden. Im Juni 1992 wurde der RFC 1652 verabschiedet, der eine wichtige Neuerung brachte: MIME (*Multipurpose Internet Mail Extensions*). Mit diesen *Mechanisms for Specifying and Describing the Format of Internet Message Bodies* war der E-Mail-Versand nicht mehr nur auf den reinen Textversand beschränkt, sondern damit wurde es möglich, auch andere Datenobjekte per E-Mail zu versenden. Insbesondere handelt es sich dabei um folgende Objekttypen:

- Anwendungsdaten *application/postscript*
- Akustikdaten *audio/basic*
- Bilder und Grafiken *image/gif* bzw. *image/jpeg*
- gemischte Datentypen *multipart/mixed*
- Text *text/plain*
- Videosequenzen *video/mpeg*

Post Office Protocol 3 (POP3)

Ein einfaches Mail-System zur Abholung von E-Mails durch den Benutzer wird durch das *Post Office Protocol 3* (POP3) realisiert. POP3 wurde als ein Mail-Standard definiert (RFC 1939 vom Mai 1996), der eine TCP-Verbindung zu einem POP3-Server aufbaut und nach Authentifizierung durch Benutzerkennung und Kennwort die auf dem Server gespeicherten E-Mails zum Benutzer, dem POP3-Client, herunterlädt. Nach Beendigung dieses Downloads werden die übertragenen Mails auf dem Server gelöscht.

Es handelt sich hierbei also um eine temporäre TCP/IP-Verbindung zwischen dem Client und dem Server, die in der Regel über bestehende Internetverbindungen durchgeführt wird. Auf dem POP3-Server (er verwendet den *well known port* 110 für sein *Listening*) werden sämtliche Benutzerdaten vorgehalten und administriert. Für jeden Benutzer gibt es ein E-Mail-Konto, das die Daten in einem einzigen Verzeichnis aufbewahrt, bis sie vom Benutzer vollständig abgeholt werden.

Im Gegensatz zum IMAP (siehe nachfolgender Abschnitt) muss der Benutzer nach Anmeldung an seinem POP3-Server sämtliche Mail-Objekte abholen. Ein partieller Download seiner E-Mails ist nicht möglich. Dies kann insbesondere bei

langsamen Verbindungen durchaus zu Übertragungszeiten führen, die mehrere Minuten bis zu Stunden umfassen. Das im Internet mittlerweile übliche Versenden umfangreicher E-Mails mit binären Anlagen (Bilder, Software, Dokumentationen usw.) trägt zu diesem ungünstigen Kommunikationsverhalten natürlich bei.

POP3-Kommandos

Die POP3-Kommunikation verwendet das *Message-Format* des *Simple Mail Transfer Protocol* (SMTP) und arbeitet daher Hand in Hand mit SMTP-Komponenten. Für die interne Kommunikation mit dem POP3-Server werden jedoch eigene Anweisungen und Befehle verwendet, die nachfolgend exemplarisch aufgeführt sind:

POP3-Kommando	Beschreibung
QUIT	Client beendet die POP3-Session.
STAT	Client führt eine Statusabfrage seines Mail-Postkorbs durch; als Antwort erhält dieser die Anzahl verfügbarer Mails und ihren Umfang in Bytes.
LIST	Je nach Parametrisierung erhält der Client eine Liste mit fortlaufender Nummerierung jeder Mail

und Angabe ihres Umfangs in Bytes; es ist jedoch auch eine gezielte Listung einzelner Mails möglich.

RETR	Mit diesem Kommando werden gezielt Mails angesprochen und vom Server zum Client übertragen.
DELE	Mit dieser Anweisung löscht der Client die Mail mit der angegebenen Mail-Nummerierung.
NOOP	<i>No operation</i> – Server bleibt passiv und führt keinerlei Aktivitäten durch.
RSET	Zum Löschen markierte Mails werden durch dieses Kommando reaktiviert. Die Markierung wird entfernt.
TOP	Server sendet nach Empfang dieses Kommandos den <i>Mail-Header</i> samt Anzahl der gewünschten Zeilen der Mail.
UIDL	<i>Unique ID Listing</i> – Client fordert – je nach Parameterübergabe – die eindeutige Identifikation einer bestimmten Mail oder eine Liste dieser Mails an. Die Gültigkeit dieser ID bezieht sich auf mehrere POP3-Sessions.
USER	Client gibt Benutzerkennung zwecks Authentifizierung gegenüber seinem Postfach an.

PASS	Client gibt Benutzerkennwort zwecks Authentifizierung gegenüber seinem Postfach an.
APOP	Verschlüsselte Authentifizierung über Benutzername und Kennwort in einer POP3-Session (nicht von allen POP3-Servern unterstützt)

Verbindungsverlauf

Bei näherer Betrachtung einer POP3-Verbindung lassen sich drei verschiedene Phasen identifizieren:

Phase 1: AUTHORIZATION

Nachdem der Client mit dem Server die POP3-Verbindung eingegangen ist, erhält dieser vom Server die folgende Meldung:

```
S: +OK POP3 server ready
```

Nun beginnt der Authentifizierungsprozess, indem der Client die Kommandos USER und PASS an den Server übermittelt.

Folgender Dialog zwischen Client (C) und Server (S) wäre möglich:

C: USER gerd

S: +OK gerd is a defined user

C: PASS geheim

S: -ERR maildrop already locked

...

C: USER gerd

S: +OK gerd is a defined user

C: PASS sehrgeheim

S: +OK gerd's maildrop has 3 messages (350 octets)

Nach erfolgreicher Authentifizierung geht die Verbindung in den Status TRANSACTION über.

Phase 2: TRANSACTION

Im *Transaction-Status* können die Kommandos STAT, LIST, RETR, DELE, NOOP, UIDL, RSET, QUIT und TOP ausgeführt werden (Erläuterungen siehe obige Tabelle). Folgende Beispiele dokumentieren das LIST- und DELE-Kommando:

```
C: LIST
```

```
S: +OK 2 messages (320 octets)
```

```
S: 1 120
```

```
S: 2 200
```

```
S: .
```

```
...
```

C: LIST 2

S: +OK 2 200

...

C: LIST 3

S: -ERR no such message, only 2 messages in mailbox

C: DELE 1

S: +OK message 1 deleted

...

C: DELE 2

S: -ERR message 2 already deleted

Durch das QUIT-Kommando wird die Phase 3, der UPDATE-Status, eingeleitet.

Phase 3: UPDATE

Die zuvor im *Transaction-Status* vor jeglichem Zugriff geschützten Ressourcen werden nun wieder freigegeben und die Verbindung zwischen Client und Server wird beendet.

Internet Message Access Protocol 4 (IMAP4)

Das *Internet Message Access Protocol 4* wird gewissermaßen als Nachfolger des POP3-Protokolls »gehandelt«. Unzulänglichkeiten insbesondere hinsichtlich fehlender Flexibilität gegenüber dem teilweisen Übertragen und Löschen von E-Mails werden hier vermieden. Außerdem ist es mit IMAP4 möglich, ein Postfach von mehreren, nicht zeitgleich betriebenen Rechnern anzusprechen. Parallel-Arbeitsplätze, also im Büro und zu Hause, können somit problemlos mit ein und demselben Postfach arbeiten, ohne dass es zu einem inkonsistenten Datenbestand kommt. Darüber hinaus bietet IMAP4 die Möglichkeit, Nachrichten in mehreren Ordnern bzw. Verzeichnissen zu archivieren und mit mehreren Mail-Servern gleichzeitig zu arbeiten.

Grundlage des IMAP4-Protokolls ist der RFC 2060 vom Dezember 1996. Die Komplexität von IMAP4 lässt eine ausführliche Beschreibung innerhalb dieses Buches nicht zu, daher erfolgt eine kurze Darstellung der funktionalen Schwerpunkte:

- Zugriff und Manipulation von Nachrichten und Ordnen auf einem Server
- Offline-Client-Synchronisation mit dem Server
- Erstellung neuer Postfächer, einschließlich Umbenennen und Löschen
- Überprüfung auf neue Nachrichten
- Selektives Übertragen von Nachrichten (bzw. Teilen dieser Nachrichten)
- Verwendung des SMTP-Protokolls

Unified Collaboration and Communication (UCC)

Vor wenigen Jahren hat man noch eine Mehrzahl multimedialer Dienste im TCP/IPbasierten Netzwerk unter der »Voice over IP«-Technologie zusammengefasst. Das reicht heute nicht mehr. Die Möglichkeiten und Erfordernisse elektronischer Kollaboration im Rahmen internationaler Zusammenarbeit (beispielsweise in virtuellen Teams) sind in den letzten Jahren wesentlich vielfältiger geworden. Es ist eine Reihe von Diensten entstanden, die man als »Unified Collaboration and Communication« (UCC) bezeichnet.

In diesem Abschnitt werden die wichtigsten Disziplinen des UCC erläutert ([Abb. 6–12](#)). Eine Detailbetrachtung muss jedoch

unterbleiben, da diese sonst den Rahmen dieses Buches sprengen würde.

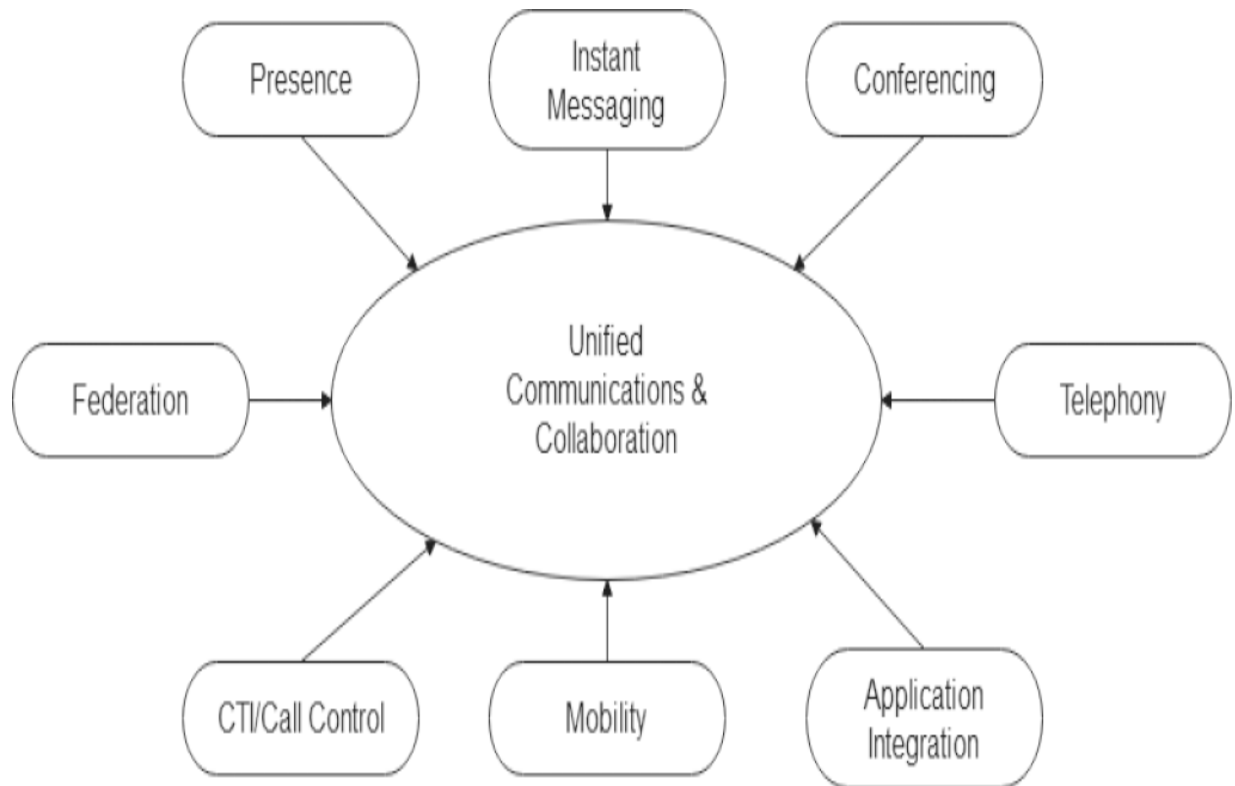


Abb. 6–12 UCC-Disziplinen

Presence Manager

Die Funktion des Presence Managers liefert eine kontinuierliche Statusinformation über Kollegen und Mitarbeiter und ihre Verfügbarkeit. In der Regel sorgt ein Ampelsystem (rot für nicht verfügbar, gelb für temporär nicht erreichbar und grün für verfügbar) dafür, dass die korrekte Information jedem UCC-Teilnehmer zur Verfügung steht (dies

erfolgt normalerweise über eine geeignete Client-Software, die von zahlreichen UCC-Providern angeboten wird).

Befindet sich ein gewünschter Kommunikationspartner beispielsweise im Status »grün«, so ist er kommunikationsbereit und für eine Textnachricht »empfänglich«. Der Status »rot« signalisiert hingegen, dass es im Moment wenig Sinn hat, einen »Chat« zu beginnen, da der Gesprächspartner voraussichtlich nicht verfügbar ist.

Die Statusinformationen werden meist dadurch ermittelt, dass die Tätigkeit eines Benutzers am eigenen Rechner »gemessen« wird. Wurde der potenzielle Gesprächspartner beispielsweise zu einer kurzen Besprechung gerufen, und seine Arbeitsstation bleibt für einige Minuten unbenutzt, geht sie automatisch in den Status »gelb« über und dokumentiert somit die temporäre Nicht-Erreichbarkeit. Hat der Benutzer seinen Status hingegen bewusst auf »nicht verfügbar« gesetzt (weil er beispielsweise weiß, dass sein Termin länger dauert), so wird dies als Status »rot« für den Gesprächspartner angezeigt.

Aber Achtung! Hier möchte in der Regel auch der Betriebsrat ein Wörtchen mitreden. Die Einführung von Presence-

Funktionen sollte daher sorgfältig im Rahmen von Betriebsvereinbarungen geregelt werden.

Instant Messaging (IM)

Das Instant Messaging ist an sich nicht neu, es hat allerdings bis vor wenigen Jahren im Geschäftsleben eher ein Schattendasein gefristet und musste der E-Mail-Technologie den Vortritt lassen. Mit zunehmender Geschwindigkeit, mit der Informationen heute in Unternehmen verfügbar gemacht werden müssen, kam die Anforderung nach einer textuellen dialogorientierten Kommunikation in Echtzeit: dem »Chat«.

Diese Kommunikationsmethode ist immer dann sinnvoll, wenn man »mal eben« eine Information benötigt und man nicht auf die Antwort einer E-Mail warten möchte (oder aus Zeitgründen nicht kann). Außerdem ist die Bereitschaft, auf eine Instant Message zwischendurch zu antworten (selbst während einer Besprechung) wesentlich größer, als ein Telefonat anzunehmen oder eine E-Mail zu schreiben. Es ist dafür allerdings unabdingbar, dass zu diesem Zweck die zuvor beschriebene »Presence«-Funktion aktiviert ist. Denn nur so kann sichergestellt werden, dass man auch tatsächlich eine sofortige Antwort erhält und ein Dialog zustande kommt.

Aktuelle Software stellt allerdings heute nicht nur die reine Chat-Funktion zur Verfügung, sondern ermöglicht im Verlaufe eines Chats auch die Übertragung von Dateien (also beispielsweise Word-Dokumente, Fotos, Grafiken usw.).

Auf die relevante Software (wie beispielsweise MS Teams von Microsoft und viele andere) soll hier nicht eingegangen werden, da dies den Rahmen des Buches sprengen würde. Im Übrigen ist der Markt hier derart schnelllebig, dass die lieferbaren Informationen für ein Buch nur eine sehr eingeschränkte Gültigkeitsdauer hätten.

Conferencing

Mit der Conferencing-Funktion wird ein sehr mächtiger Dienst für UCC erschlossen. Er umfasst normalerweise Audio- und Videofunktionen sowie die Möglichkeit des »Desktop Sharing« (also die gemeinsame Nutzung eines Desktops; z.B. von einem Gesprächspartner, der gerade eine Präsentation einer Gruppe von Teilnehmern vorführen möchte, oder wenn man gerade an einem gemeinsamen Dokument arbeiten soll).

Eine Besonderheit beim Conferencing ist der recht hohe Bedarf an Netzwerkbandbreite, dessen Realisierung heutzutage bei den zur Verfügung stehenden hohen Leitungskapazitäten kein Problem darstellen sollte.

Das Conferencing für die Arbeit in Teams – insbesondere in internationalen virtuellen Teams – ist unabdingbare Voraussetzung für eine effektive Zusammenarbeit. Es kann außerdem dazu beitragen, nicht unbedingt erforderliche Reisen zu sparen, da umfängliche Videokonferenzen einschließlich der gemeinsamen Arbeit mit Dokumenten manch eine Präsenzkonferenz ersetzen können. Außerdem erleichtert dieser Dienst das Arbeiten im Homeoffice und schafft somit eine wichtige Grundlage für eine in Zukunft zu erwartende neue von Präsenz in Büros unabhängige Arbeitsweise.

Telephony

Die Integration der klassischen Telefonie hat sich im UCC-Umfeld erst relativ spät vollzogen. Die zu ihrer Implementierung erforderlichen Technologien waren entweder noch nicht voll ausgereift oder es fehlten einfach die benötigten Bandbreiten auch über lokale Netzwerke hinaus. In den letzten Jahren hat sich dies geändert.

Insbesondere sind es die folgenden Kriterien, die diese Entwicklung begünstigt haben:

- Optimierung von Audio-Codecs
- Traffic-Priorisierung in Netzwerken (*Quality of Service*)

- Einführung von Quality of Service für Echtzeitanwendungen (Voice, Video)
- Erschwingliche Breitbandverbindungen
- Schnittstellen zur Integration alter Analogsysteme
- Weiterentwicklung des Session Initiation Protocols (SIP)

Seit Erscheinen des Ur-RFC für das Session Initiation Protocol hat sich in diesem Bereich eine Menge getan. Das Anwendungsprotokoll für die Kommunikation zwischen Partnern (Punkt-zu-Punkt oder auch Multi-Punkt) regelt ja bekanntlich ihr Verhalten bei Internettelefonie, Multimedia-Datenströmen und Multimedia-Konferenzen (auch Webkonferenzen oder Collaboration genannt). Diese Disziplinen haben in den letzten Jahren eine Vielzahl von Aktualisierungen erhalten, die in ebenso zahlreichen RFCs beschrieben sind. Der zuletzt veröffentlichte RFC trägt die Nummer 9027 vom Juni 2021 und hat den Titel »Assertion Values for Resource Priority Header and SIP Priority Header Claims in Support of Emergency Services Networks«.

Heute haben die meisten wichtigen UCC-Provider auch Telephony in ihrem Portfolio, sodass UCC nunmehr auch als Gesamtpaket Einzug in die Unternehmensinfrastruktur gehalten hat. Die Tage konventioneller Telefonanlagen sind gezählt.

Application Integration

Die Integration von Sprache in zentrale Anwendungen (beispielsweise SAP) hat sich als äußerst effektiv erwiesen. So wurden Applikationen entwickelt, aus denen heraus beispielsweise direkt telefoniert werden kann (MS Outlook, SAP-Phone usw.) oder auch Zusatzfunktionen zur Verfügung gestellt werden können (z.B. automatische Anzeige von Kundendaten bei eingehendem Anruf). Zahlreiche weitere Hersteller sorgen dafür, dass der Wettbewerb immer neuere, aber auch stabilere Systeme auf den Markt bringt (Cisco, Avaya, XPhone, Unify, Microsoft usw.).

Mobility

Die Erweiterung des UCC um Mobility-Lösungen ist ein konsequenter Schritt, der auch die Welt von Smartphones, Tablets und sonstigen mobilen Clients einbezieht. Services wie Bring-Your-Own-Device (BYOD) werden ebenso integriert wie die Nutzung technologieübergreifender Telefonnummern (keine separaten Telefonnummern für Fest- und Mobilnetze).

Unter dem Begriff »Fixed Mobile Convergence« (FMC) hat sich vor einigen Jahren eine Disziplin herausgebildet, die für eine Kombination der Technologien WiFi, VoIP, Mobilnetze, klassische Telefonanlagen (PBX = *Private Branch Exchange*) und

dem Internet steht. Sie ist wesentlicher Bestandteil der UCC-Disziplin »Mobility«.

Deutlich gestiegene Bandbreiten im Umfeld der heutigen Mobilnetze (z.B. 4G/5G) tragen ferner dazu bei, dass diese Technologie immer mehr an Bedeutung gewinnt. Bandbreiten im Consumer-Bereich von 50 Mbit/s gehören bereits zum Standardtarif – höhere Bandbreiten von 100 bis 500 Mbit/s – sogar bis 1 GBit/s – sind ebenfalls möglich. Der vollständigen Integration mobiler Datendienste in Festnetze oder ins LAN/WAN-Umfeld steht daher nichts mehr entgegen.

CTI und Call Control

Unter den Disziplinen CTI (*Computer Telephony Integration*) und Call Control versteht man die wesentlichen Elemente zur Verbindung von Anwendungen mit der Telefonanlage. Dieses Zusammenspiel hat bereits in früheren Tagen dafür gesorgt, dass man integrierte Dienste nutzen konnte, ohne bereits von UCC zu sprechen. Verschiedene Hersteller verbinden ihre serverbasierten Systeme mit der Telefonanlage und können somit den Anwendern eine Vielzahl von Mehrwerten zur Verfügung stellen (eine sehr populäre Lösung sind Call-Center-Implementierungen).

Federation

Letztlich gehört die unternehmensübergreifende UCC-Implementierung zur Königsdisziplin, denn so gut man auch die interne Kommunikation im Rahmen von UCC optimiert haben mag, so stellt die einzubeziehende Kommunikation nach außen hin eine der wichtigsten Disziplinen dar. Sie wird als »Federation« bezeichnet.

Dabei geht es nicht unbedingt um eine 100-Prozent-Integration zweier oder mehrerer Unternehmen und ihrer UCC-Lösungen, sondern meist um Teilbereiche. So kann es als außerordentlich sinnvoll und effektiv angesehen werden, wenn zwei Unternehmen beispielsweise ihre Telephony- und Conferencing-Disziplinen miteinander verbinden und darüber gemeinsam kommunizieren. Die simultane Arbeit an gemeinsamen Projekten durch Bearbeitung ein und desselben Dokuments ist dann problemlos möglich.

Lightweight Directory Access Protocol (LDAP)

Zur besseren Übersicht und Verarbeitung werden Informationen im Allgemeinen hierarchisch strukturiert. Dies gilt für den privaten Bereich, aber auch besonders für die Ressourcen in Unternehmen. Um ein umfassendes Informationsprofil eines Unternehmens zu erstellen, sind

beispielsweise Daten über seine Mitarbeiter und Produkte erforderlich. Der einzelne Mitarbeiter muss dann durch Namen, Anschrift, Telefonnummer, E-Mail-Adresse und Zuständigkeit genau beschrieben werden.

Detaillierte Adresslisten innerhalb von E-Mail-Systemen repräsentieren hier oft diejenigen Verzeichnisse, die über das *Lightweight Directory Access Protocol* (LDAP) administriert werden können.

Konzeption

Ursprünglich entstammt LDAP dem OSI-Standard X.500, dessen funktional sehr umfangreiches Zugriffsprotokoll DAP (*Directory Access Protocol*) in der Praxis kaum zum Einsatz kam, denn aufgrund der Komplexität war es für eine Implementierung in typischen Clients (Personalcomputern) nicht geeignet. Dies änderte sich erst, als 1994 die erste Version des LDAP veröffentlicht wurde, um durch eine vereinfachte Philosophie jedem Rechnertyp den Zugriff auf X.500-basierte Datenbestände zu ermöglichen.

Im Gegensatz zum X.500 basiert LDAP auf dem *Transmission Control Protocol* (TCP) und verwendet serverseitig den Port 389. Es kann daher von allen TCP/IPbasierten Rechnern benutzt

werden, sofern die erforderlichen LDAP-Applikationen implementiert sind (z.B. E-Mail-Clients oder Browser).

Die Definition der Datenstruktur erfolgt durch die Spezifikation von *Objektklassen* (fest vorgegeben oder individuell), *Attributen* und *Werten*. Die Objektklasse *Organizational Unit* ist beispielsweise zwingend erforderlich und wird stets zu Beginn einer LDAP-Datenstruktur formuliert. Hinter dem sogenannten *distinguished name* (dn) verbirgt sich zumeist die eindeutige Internetdomäne des Unternehmens bzw. der Organisation, aufgeteilt in sogenannte *domain components* (dc).

Grundsätzlich können LDAP-Daten interaktiv oder als Eingabedatei zur Verfügung gestellt werden. Infolge der sehr aufwendigen Eingabedialoge werden zumeist Eingabedateien erstellt, die dann beim Start des LDAP-Serverprozesses eingelesen werden.

Folgendes Beispiel zeigt eine LDAP-Eingabedatei:

```
dn: dc=meier, dc=de

objectclass: organisation
```

objectclass: top

o: Karl Meier GmbH

l: Muenchen

postal code: 85677

streetAddress: Lange Str. 244

dn: ou=development, dc=meier, dc=de

objectclass: organizationalUnit

ou: development

description: Entwicklungsabteilung

telephonenumber: 089-2355-45

emailaddress: development@meier.de

dn: ou=humanresources, dc=meier, dc=de

```
objectclass: organizationalUnit  
  
ou: humanresources  
  
description: Personalabteilung  
  
telephonenumber: 089-2355-46  
  
emailaddress: hr@meier.de
```

In diesem Beispiel wurde das Unternehmen *Karl Meier GmbH* inklusive Anschrift mit zwei Abteilungen, ihren Telefonnummern und E-Mail-Adressen definiert. In ähnlicher Art und Weise könnten beispielsweise Einträge zu den Mitarbeitern oder anderen Ressourcen des Unternehmens angelegt werden.

Application Programming Interface (API)

Für eine Integration von LDAP-Zugriffen in eigene Programme wird eine entsprechende Schnittstelle (API) zur Verfügung gestellt. Eine beispielhafte Übersicht der verwendbaren Funktionen ist der nachfolgenden Tabelle zu entnehmen.

LDAP-Operation	Beschreibung
<code>ldap_open()</code>	Verbindung zum LDAP-Server wird aufgebaut.
<code>ldap_bind()</code>	Verzeichnis-Authentifizierung
<code>ldap_unbind()</code>	Verbindung zum LDAP-Server wird beendet.
<code>ldap_search()</code>	Suchen im LDAP-Verzeichnis
<code>ldap_modify()</code>	Verzeichniseintrag verändern
<code>ldap_add()</code>	Verzeichniseintrag hinzufügen
<code>ldap_delete()</code>	Verzeichniseintrag löschen
<code>ldap_abandon()</code>	Verbindung zum Server wird unterbrochen.

HINWEIS

Der zunehmende Bedarf an einem zentralen Verzeichnisdienst hat in Unternehmen dazu geführt, dass LDAP in bereits bestehende Serverdienste wie Microsofts *Active Directory Services* oder andere zentrale Verzeichnisdienste integriert worden ist.

NFS

Das *Network File System* (NFS) ist eine Entwicklung von Sun Microsystems und stellt ein Dateisystem dar, das unabhängig von Hardware, Betriebssystem und Netzwerk eingesetzt werden kann. Es verbindet heterogene Netzwerk- und

Rechnerarchitekturen über die Implementierung eines Dateisystems. Diese Unabhängigkeit wird erreicht, da auf Basis von UDP das Konzept des *Remote Procedure Call* (RPC) und die *External Data Representation* (XDR) verwendet werden.

NFS, oft auch als ONC (*Open Network Computing*) bezeichnet, beruht auf dem vollständigen Protokollstack des *Internet Protocol* (siehe [Abb. 6–13](#)), wobei der Schwerpunkt eindeutig auf UDP als Basisprotokoll liegt. Einige Implementierungen verwenden auch TCP. Die Gründe liegen meist in der Absicht, NFS auch über WAN-Netzwerke zu betreiben. Dies ist über UDP meist nur dann zufriedenstellend möglich, wenn der Datagramm-Verlust in Grenzen gehalten werden kann (also im LAN).

Die Funktionalität von NFS wird in den drei oberen Protokollschichten abgebildet. In Schicht 5 stellt das RPC Funktionen zur Verfügung, die ein zur Auswertung in höheren Schichten standardisiertes *Message-Format* definieren und Interprozess-Kommunikationstechniken ermöglichen. Ferner wird durch das sogenannte *Caller-Server-Prinzip* der Eindruck erweckt, als liefе ein an einen entfernten Serverprozess irgendwo im Netzwerk delegierter Ausführungsauftrag lokal auf dem eigenen Rechner.

- 7 NFS
- 6 XDR
- 5 RPC
- 4 UDP - TCP
- 3 Internet Protocol
- 2 Data Link Layer
- 1 Physical Layer

Abb. 6–13 NFS-Protokollarchitektur

Remote Procedure Calls (Layer 5)

Das Konzept der Remote Procedure Calls (RPC) bietet die Möglichkeit einer umfassenden Netzwerkkommunikation unter Einbeziehung aller Rechnerressourcen zum *Distributed Computing* (siehe [Abb. 6–14](#)). Eine gezielte Verteilung bestimmter Aufgaben innerhalb eines Netzwerks (auch über verschiedene Topologien) führt zu einer Optimierung der verfügbaren Rechnerressourcen. Sind Rechner mit einer komplexen Aufgabe zu einem bestimmten Zeitpunkt überfordert, so kann die Ausführung der erforderlichen Verarbeitung an andere Rechner mit geringerer Belastung delegiert werden – oder aber Rechner mit einer höheren Prozessorleistung bzw. weitere Prozessoren (in Multiprozessormaschinen) können für diese Aufgaben

herangezogen werden. Alle Netzwerkressourcen avancieren zu einem gigantischen Rechnerkomplex.

HINWEIS

Eine gezielte Delegation bestimmter Teilaufgaben kann auch zu einer Minimierung der Netzwerkbelastung führen. Werden Daten von vornherein dort verarbeitet, wo sie auch für eine weitere Verarbeitung benötigt werden, so entfällt ihr Transport über das Netzwerk.

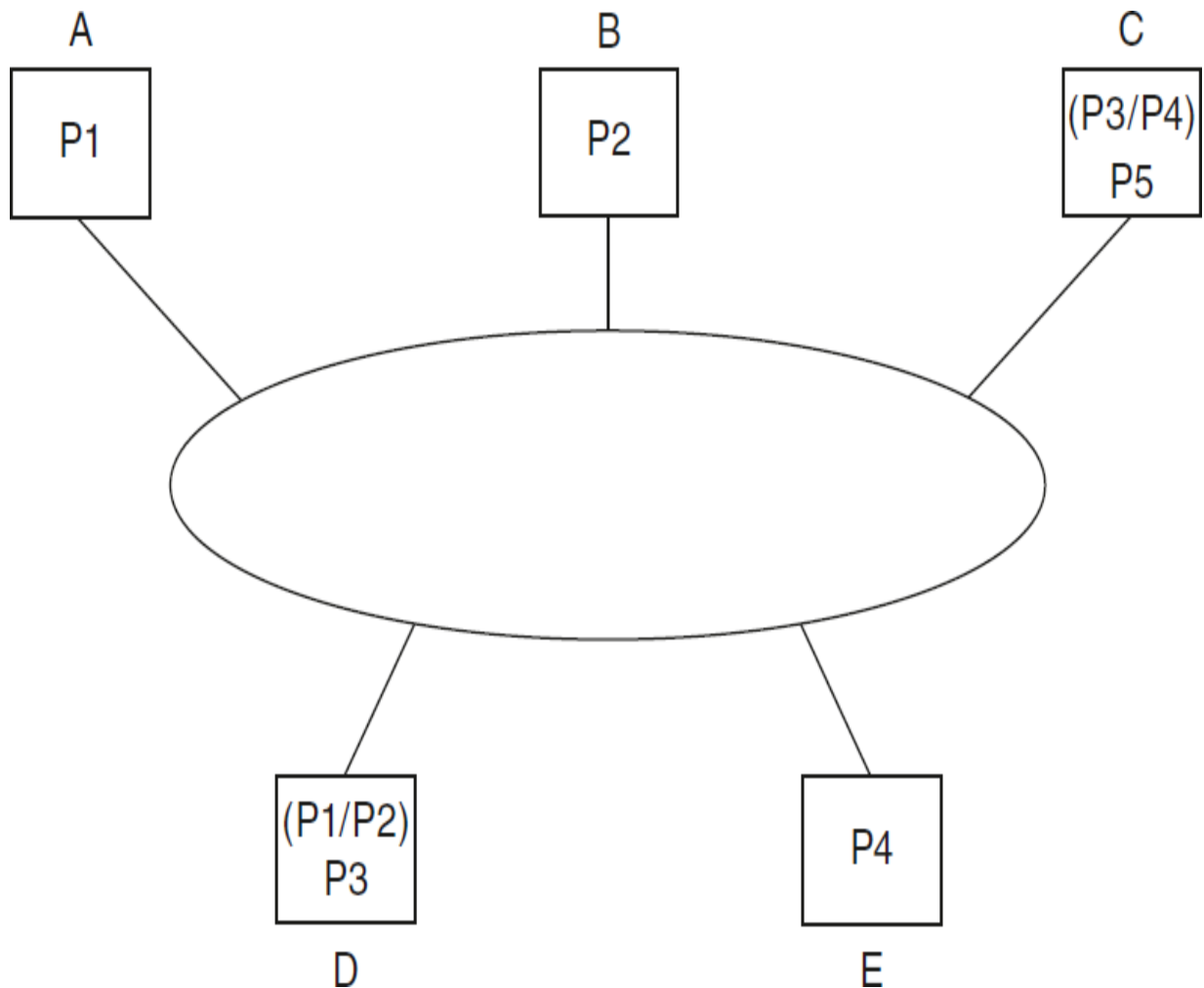


Abb. 6–14 *Distributed Computing durch RPC*

Für die Ermittlung eines vorgegebenen Verarbeitungsergebnisses stehen verschiedene Rechner unterschiedlichen Typs innerhalb eines Netzwerks zur Verfügung. Rechner A und Rechner B sind äußerst schnelle Systeme (Personalcomputer), Rechner D und E sind UNIX-Workstations, die für Datenbankoperationen optimiert sind. Rechner C ist schließlich eine schnelle Grafikmaschine mit Hardwarekomponenten für die optimierte Präsentation von

Ergebnissen in Form komplexer Grafiken. Die verschiedenartigen Aufgaben werden nun so verteilt, dass für eine Vorabberechnung umfangreicher Matrizen Prozess P1 und P2 auf den beiden PCs vorgenommen wird. Die Ergebnisse werden Rechner D zur Verfügung gestellt, der seinerseits Datenbankoperationen vornimmt. Weiterführende Datenbankoperationen werden parallel (oder anschließend) von Rechner E durchgeführt. Das Ergebnis wird letztlich dem Grafikrechner C übermittelt, damit eine geeignete grafische Darstellung erfolgen kann. Jeder Rechner führt genau die Aufgaben aus, die seinem Leistungsprofil am ehesten entsprechen. Können Verarbeitungen parallel ausgeführt werden, so nutzt man auch diese Möglichkeit, sofern ein entsprechender Rechner seine Kapazitäten zur Verfügung stellen kann.

Die Kommunikation zwischen zwei *Remote Procedures* erfordert neben der Verwendung spezieller Funktionen, die in gesonderten Bibliotheken innerhalb von APIs (*Application Programming Interface*) zur Verfügung stehen, auch die Definition geeigneter Protokollheader für einen *RPC-Request* und einen *RPC-Reply*.

External Data Representation (XDR)

In Schicht 6 wird über die *External Data Representation* (XDR) durch eine einheitliche Beschreibung von Datentypen und die Existenz einer eigenen Datenbeschreibungssprache ein in jeder Hinsicht unabhängiger Datenaustausch ermöglicht. Die XDR liefert somit eine unmissverständliche Präsentation auf Kommunikationsebene 6. Hier wird eindeutig festgelegt, wie ein Rechner (unabhängig davon, welcher Architektur er angehört oder welches Betriebssystem er verwendet) bzw. eine Anwendung die empfangenen Daten zu interpretieren hat.

Bevor der Client die Daten an das Netzwerk übergibt, müssen diese festgelegten Definitionen entsprechen. Durch diese Festschreibung infrage kommender Datenobjekte ist es gleichgültig, welchen Ursprungs die kommunizierenden Prozesse sind. Werden die getroffenen Vereinbarungen nach dem XDR-Konzept eingehalten, kann es zu falschen Interpretationen des Datenstroms nicht kommen.

Eine weitere Standardisierung besteht darin, dass die Datendarstellung in einem Vielfachen von 4-Byte-Blöcken erfolgt. Ist das tatsächliche Volumen geringer, so werden binäre Nullen als »Füller« verwendet. Die vom NFS-Client vorgenommene Formatierung gemäß XDR wird nach

Übertragung und Empfang durch den NFS-Serverprozess wieder aufgelöst und an das lokale Betriebssystem weitergereicht.

Prozeduren und Anweisungen

NFS selbst bietet eine Vielzahl von Prozeduren, die für ein komfortables und leistungsfähiges Datei-Handling über Netzwerkgrenzen hinaus notwendig sind. Ein Auszug findet sich in der nachfolgenden Tabelle.

NFS-Prozedur	Prozedur-Nummer	Beschreibung
NULL	0	keine Aktivitäten, lediglich Lieferung eines Returncodes
GETATTR	1	Dateiattribute abfragen
SETATTR	2	Dateiattribute setzen
READ	6	aus einer Datei x Bytes lesen
CACHE	7	in den Cache-Speicher schreiben
WRITE	8	in eine Datei x Bytes schreiben
CREATE	9	eine Datei generieren
DELETE	10	eine Datei löschen
RENAME	11	eine Datei umbenennen
LINK	12	eine Dateiverbindung herstellen
MKDIR	14	ein Verzeichnis erstellen

RMDIR	15	ein Verzeichnis löschen
STATFSRES	17	Attribute eines Dateisystems lesen

Der Auslöser für die Verfügbarkeit der NFS-Funktionalität ist der Befehl MOUNT. Wird dieses Kommando erfolgreich ausgeführt, so stehen die entfernten Datei- bzw. Dateisystemressourcen auf dem lokalen Host zur Verfügung und können mit den vorgenannten Prozeduren ebenfalls verwaltet werden. Die Anweisung UNMOUNT hebt diesen Zustand wieder auf.

Charakteristisch für die Client-Server-Beziehung im NFS-Umfeld ist ihre Zustandslosigkeit (*Stateless Relationship*). Synchronisierungsverfahren zur Ermittlung des jeweils aktuellen Status von Client und Server sind nicht erforderlich, da bei einem *Command-Request* durch den Client unmittelbar nach Rückgabe eines positiven *Return-Code* der Client davon ausgehen kann, dass die gewünschten Operationen bereits erfolgreich durch den Server vorgenommen worden sind. Wird beispielsweise ein DELETE-Kommando abgesetzt und der Befehl vom Server positiv quittiert, so wurde auf dem entfernten Rechner tatsächlich physisch der Löschvorgang ausgeführt. Eine Verständigung zwischen Client und Server

über mögliche Störungen oder Probleme während der Befehlsausführung erfolgt nicht.

Seit dem RFC 3530 vom April 2003 liegt das NFS-System erstmalig in der Version 4 vor. Sein letztes Update zur Version 4.1 erfolgte im RFC 8881 vom August 2020, in dem primär Anpassungen zum Multi-Server Namespace vorgenommen wurden.

Network Information Services (NIS) – YELLOW PAGES

Für die Verwaltung der unter NFS möglichen Operationen, Zugriffsrechte und Sicherheitsobjekte (Kennungen, Kennwörter usw.) wurde von Sun Microsystems ein Informationssystem entwickelt: YELLOW PAGES. Das sich mittlerweile zum Managementsystem ausweitende YP wurde zwischenzeitlich in *Network Information Services* (NIS) umbenannt. Dieses Konzept soll dafür sorgen, dass durch eine zentrale Verwaltung dezentral organisierter Systeminformationen eine vereinfachte Systemadministration ermöglicht werden kann. Kommunikationsbasis auch hierfür sind die *Remote Procedure Calls* von Sun.

NIS wird im Wesentlichen durch eine dezentrale NIS-Datenbank, einen NIS-Master-Server, einen NIS-Slave-Server, eine NIS-Domäne und die NIS-Clients gebildet:

- *NIS-Datenbank*

Die durch zentrale Kontrolllisten für Kennungen, Kennwörter und Host-Namen entstehende Datenbank kann quasi als große */etc/passwd-Datei* für ein NIS-Netzwerk betrachtet werden.

- *NIS-Master-Server*

Dieser Server verwaltet die NIS-Datenbank für eine definierte NIS-Domäne.

- *NIS-Slave-Server*

Dient als Ausfallsicherung für den NIS-Master-Server. Die NIS-Datenbank muss daher auf dem vorgesehenen *NIS-Slave-Server* in Kopie vorgehalten werden, um Datenkonsistenz zu gewährleisten. Der *Slave* übernimmt auch dann die NIS-Datenbank, wenn er den Master entlasten soll.

- *NIS-Domäne*

Eine Gruppe von Rechnern, die in einer NIS-Datenbank abgebildet werden.

- *NIS-Client*

Rechner, die beim NIS-Server Informationen aus der NIS-Datenbank beziehen können. Modifikationen der Datenbank können jedoch nur zentral vom Server vorgenommen werden.

Das NIS-Prinzip erfordert, ausgehend vom Client, folgende Fragestellung:

- Wo befindet sich der NIS-Server?
- Wie lautet der Name des NIS-Servers bzw. wie kann der Server adressiert werden?
- Wo befindet sich die gewünschte Information?

Diese drei Phasen der Akquisition werden durch die Kommandos YPBIND, YPSERV und YPCAT implementiert. Voraussetzung für eine Kommunikation sind aktive YPBIND-Hintergrundprozesse auf jedem Client-System. Ein *Broadcast* ermittelt die IP-Adresse des NIS-Servers und speichert diese. Ist der NIS-Server nicht mehr aktiv, wird mit dem gleichen Verfahren ein neuer Server gesucht. Grundsätzlich wird der Server akzeptiert, der zuerst auf den *Broadcast* antwortet. Auf diesem Wege ist somit auch Lastverteilung möglich. Der YPSERV-Prozess auf dem NIS-Server stellt anschließend die gewünschten Informationen zur Verfügung, die vom Client per YPCAT-Kommando schließlich als NIS-Listen übernommen werden.

Da NIS praktisch keinerlei Sicherheitsfunktionen enthält, wird es heute kaum noch eingesetzt. Die von SUN weiterentwickelte Variante NIS+ hat sich nicht durchgesetzt. Stattdessen wurden LDAP und Kerberos in zahlreichen Implementierungen verwendet, die auch plattformübergreifend betrieben werden können.

Kerberos

Kerberos repräsentiert ein Sicherheitssystem auf der Basis von Verschlüsselungstechniken und steht innerhalb eines Netzwerks für die Authentifizierung und Autorisierung zwischen Benutzern und Servern zur Verfügung. Es ermöglicht die Bestimmung der Echtheit eines Benutzers gegenüber dem Server und umgekehrt. Ein Transport von Kennwörtern über das Netz wird allerdings nicht vorgenommen.

Für die Realisierung einer Kerberos-Implementierung sind folgende Voraussetzungen zu erfüllen:

- Auf allen beteiligten Rechnern muss eine synchrone Zeit verfügbar sein,
- alle Server müssen physisch gesichert sein,
- ein Remote Login (*rlogin*) ist zu den Servern nicht erlaubt,
- sonstige Sicherheitsaspekte (z.B. Passwörter) müssen bereits vorhanden sein.

Bei den Kerberos-Komponenten handelt es sich im Einzelnen um folgende:

- Kerberos-Datenbank
- Kerberos-Authentifizierungsserver (KAS)
- Kerberos Ticket Granting Server (KTGS)

- Kerberos-Kommandos innerhalb von Anwendungsprogrammen

Ein Authentifizierungsprozess mit Kerberos vollzieht sich in fünf Phasen (siehe hierzu auch [Abb. 6–15](#)):

- *PHASE 1 – Authentifizierungs-Request des Users beim KAS*

Nach der Präsentation des *Login-Prompt* gibt der Benutzer seine Benutzer-Identifikation ein. Die Kerberos-Routinen im lokalen Rechner versehen diese mit einem Zeitstempel (*Time Stamp*), der Gültigkeitsdauer sowie dem Namen des Ticket Granting Server und schicken diese Message als Request zum KAS. Dieser Vorgang wird auch als KINIT bezeichnet.

- *PHASE 2 – Ausgabe eines Ticket Granting Ticket (TGT)*

Der KAS prüft in Phase 2 die empfangene User-ID und alle anderen Informationen in seiner Kerberos-Datenbank. Anschließend gibt er in verschlüsselter Form diese Informationen zurück und ergänzt (natürlich auch verschlüsselt) das Kennwort und den Namen des *Ticket Granting Server* (TGS). Nach Empfang dieses vorläufigen Ticket Granting Ticket ergeht auf dem lokalen (Client-) Rechner an den Benutzer die Aufforderung, das Kennwort einzugeben. Mit diesem Kennwort werden alle verschlüsselten Daten, die vom KAS empfangen wurden, entschlüsselt. Wird Übereinstimmung erzielt, so wird das

vorläufige *Ticket Granting Ticket* zu einem endgültigen TGT, das nun beim Client verbleiben darf. Außerdem wird dem Client ein sogenannter temporärer *Session Key* für den *Request* beim Ticket Granting Server zur Verfügung gestellt.

- *PHASE 3 – Service Ticket Request zum TGS*

In Phase 3 fordert der Client ein mit dem aus der vorigen Phase erhaltenen temporären *Session Key* verschlüsseltes Ticket beim TGS an. Er übermittelt dabei dem TGS den Servicenamen, das TGT, eine neue Client-Authentifizierung, den Zeitstempel (*Time Stamp*) und die Gültigkeitsdauer. Mit diesen Informationen wird die Überprüfung durch den TGS vorgenommen, ob der anfordernde Client die gewünschte Ressource verwenden darf.

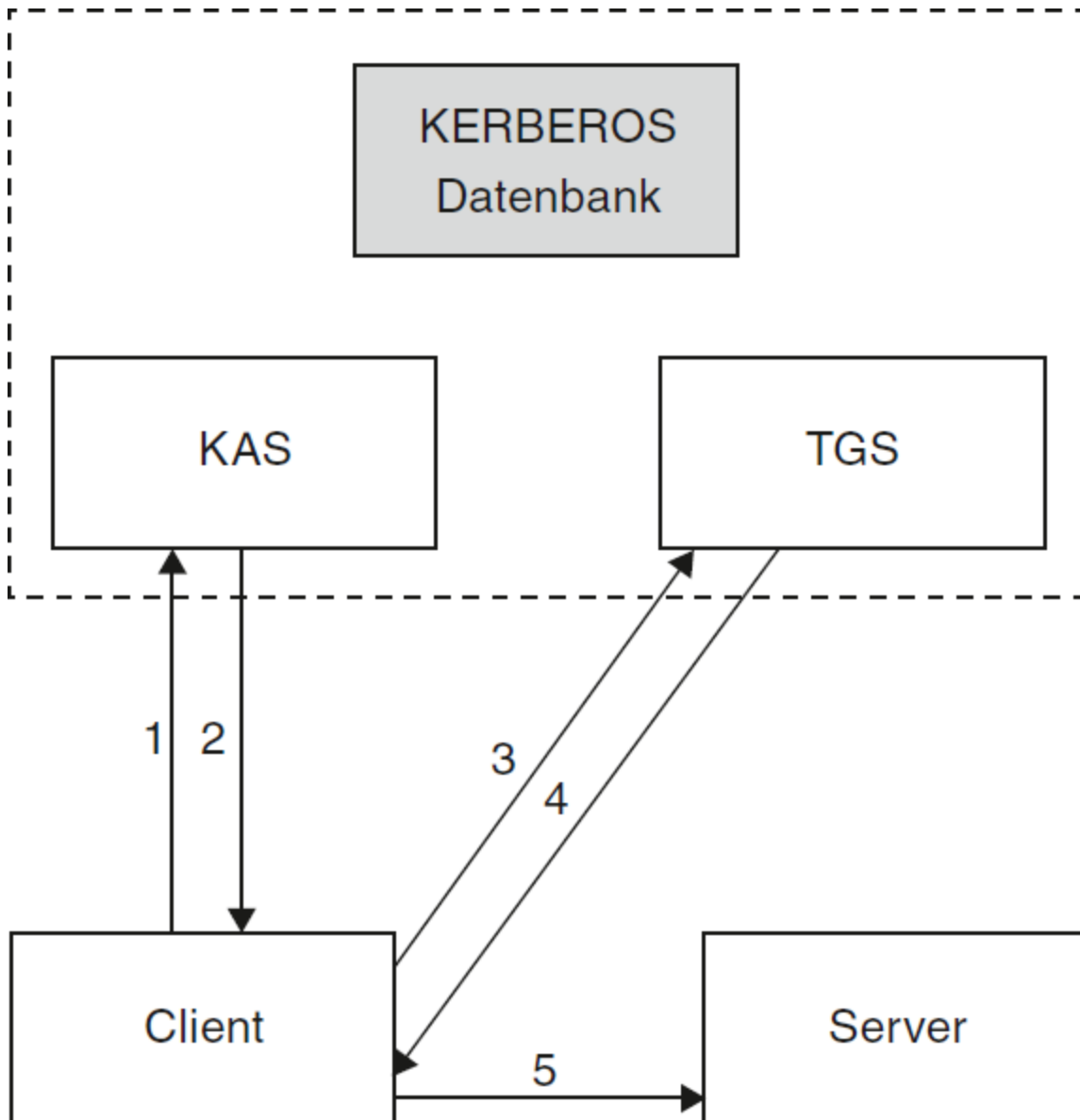


Abb. 6–15 Prinzip der Kerberos-Authentifizierung

- *PHASE 4 – TGS schickt Service Ticket zum Client*

Fällt das Prüfungsergebnis des TGS positiv aus, so übergibt dieser dem Client ein Ticket mit dem Namen des gewünschten Client-Programms, dem Vermerk »Client darf das Programm ausführen«, dem *Time Stamp* und der

Gültigkeitsdauer. Der Client entschlüsselt die empfangene Message und erhält vom TGS einen *Session Key* sowie ein Client Ticket. Wiederum erfolgt ein Vergleich, der bei Übereinstimmung dazu führt, dass der Client das erhaltene Service Ticket behält.

- *PHASE 5 – Client sendet Authentication Request zum Anwendungsserver*

Mit dem erhaltenen Service Ticket und der Client-Authentifizierung wird der gewünschte Service zum Anwendungsserver übertragen. Dieser entschlüsselt die Message mit seinem privaten *Service Key* (der lediglich ihm selbst und der Kerberos-Datenbank bekannt ist). Anschließend verwendet er den in der empfangenen Message enthaltenen *Session Key*, entschlüsselt den *Authenticator* und führt anschließend den gleichen Vergleich durch, der bereits in Phase 4 beschrieben wurde. Endet dieser Vergleich ebenfalls positiv, so wird das gewünschte Programm gestartet. Andernfalls erfolgt ein unverzüglicher Abbruch.

Die Kerberos-Datenbank enthält:

- Benutzer und Service (*Principal*)
- Privater Schlüssel für den *Principal*
- Ablaufdatum der Identität

- Datum der letzten Satzänderung
- Identität des *Principal*, der den betreffenden Satz zuletzt geändert hat
- Maximale Gültigkeit der diesem *Principal* übergebenen Tickets

Simple Network Management Protocol (SNMP)

Das *Simple Network Management Protocol* (SNMP) gehört ebenfalls zur Gruppe der TCP/IP-Anwendungsprotokolle, wobei SNMP eine gewisse Komplexität zugeschrieben werden kann.

Bislang wurden zahlreiche Netzwerkprotokolle vorgestellt, die im Wesentlichen die Aufgabe hatten, eine Übertragung von Nutzdaten zu gewährleisten. SNMP hingegen ist in erster Linie für den Transport von Kontrolldaten zuständig, um den Fluss von Managementinformationen, Status- und Statistikdaten zwischen einzelnen Netzwerkkomponenten und einem Managementsystem zu ermöglichen. Somit geht die Aufgabenstellung dieses Protokolls in eine völlig andere Richtung.

Die Hauptaufgabe im Netzwerkmanagement liegt nicht nur in der Beseitigung aktueller Störungen, sondern auch in der Beobachtung signifikanter Messstellen im Netzwerk, um

Symptome zu ermitteln, die bevorstehende Netzwerkprobleme ankündigen. Diese Vorsorge wird allgemein als *proaktives Monitoring* bezeichnet, womit die Überprüfung von Störungen und deren Eliminierung (Analyse und *Operating*) ermöglicht werden. Es handelt sich hierbei um klassische Disziplinen des Netzwerkmanagements.

In ihrer Zusammenfassung und Erweiterung um Aufgaben der Konfiguration und Leistungssteigerung ergibt sich ein umfassendes *Systemmanagement* mit folgenden Teildisziplinen:

- Configuration Management
 - Erkennung einzelner Netzwerkkomponenten
 - Ermittlung der gesamten Netztopologie
- Performance Management
 - Erkennung signifikanter Problemsituationen
 - Alert- bzw. Alarmmanagement
 - Netzwerkanalyse und -Monitoring
 - Fehlerbearbeitung
- Change Management
 - Verteilung von Software (Installation und Konfiguration)
 - Modifikation von Netztopologien (*Move-Management* = Verwaltung von Umzügen)
 - Durchführung von Planungsaufgaben
- Operations Management

- Durchführung des klassischen Operatings
- Trouble Ticketing (Bildung einer Problemdatenbank)
- Business Management
 - Gewährung ausreichender Zugangssicherheit
 - Datensicherheit

In fast allen genannten Disziplinen wird SNMP als Netzwerkprotokoll eingesetzt. Dabei wurde die Situation im Netzwerkmanagement von folgenden Entwicklungen entscheidend beeinflusst:

- Vielzahl von Netzwerkkomponenten und eine hohe Zahl von Netzwerkprotokollen
- Kostenreduktion durch zentrales Netzwerkmanagement
- Hardware verschiedener Hersteller (Heterogenität) erschwert das zentrale Management.
- Vergrößerung der Netzwerke von kleinen LANs zu umfangreichen LAN-WAN-Konstruktionen
- Einführung von Switches auf verschiedenen ISO/OSI-Layern (Layer 2 bis Layer 4)

Über das eigentliche SNMP-Protokoll hinaus werden nachfolgend die SNMP-Softwarearchitektur, die notwendigen Netzwerkkomponenten im Managementprozess und die Darstellungsverfahren für die erforderliche

Informationsstruktur erläutert. Abschließend erfolgt eine Gegenüberstellung mit den SNMP-Weiterentwicklungen SNMPv2 und SNMPv3 sowie eine Übersicht einiger wichtiger Produkte, die das SNMP-Protokoll für den Transport ihrer Managementinformationen verwenden.

SNMP und CMOT – zwei Entwicklungsrichtungen

Mitte der 80er Jahre des vorigen Jahrhunderts wurde infolge der enormen Ausdehnung der Internetnetzwerke mit dem damit verbundenen drastischen Anstieg an Netzwerkknoten das Problem der Netzwerkbetreuung offenkundig. Man registrierte pro Jahr eine Verdopplung der angebundenen Netzwerke und die breite Streuung verschiedener Hersteller. Geeignete Hilfsmittel für das Netzwerkmanagement fehlten jedoch. Als Resultat einer intensiven Zusammenarbeit zwischen Firmen und den Betreibern einiger akademisch orientierter Netzwerke wurde 1987 das Konzept für das erste Managementprotokoll SGMP (*Simple Gateway Monitoring Protocol*) entwickelt.

Ein Jahr später kam es nach diversen Zusammenkünften innerhalb des IAB zu einer Strategie, die als kurzfristige Lösung dem SGMP zusprach, jedoch als langfristige Konzeption das CMOT (*Common Management Information Services Over TCP/IP*)

vorstellte. Es wurden zwei Arbeitsgruppen gebildet, die für eine rasche Realisierung beider Protokolle verantwortlich waren. Unter Verwendung des CMIP (*Common Management Information Protocol*) und der CMIS (Common Management Information Services) wurde mit CMOT versucht, ein reinrassiges OSI-Protokoll zu entwickeln. Allerdings hat gerade diese OSI-basierende Komplexität des Protokolls dazu geführt, dass sich die große Mehrheit aller Hersteller und potenziellen Nutzer von CMOT zurückgezogen haben, da die Kosten für seine Implementierung nicht unerheblich sind. Diese mangelnde Akzeptanz hat dazu geführt, dass CMOT vom IAB immer noch im Status des *Draft Standard* gehalten wird. Nachdem das Unternehmen IBM nach anfänglichem CMIP-Engagement schließlich im Herbst 1994 seine Aktivitäten vollends einstellte, ergaben sich für die CMIP-CMIS-CMOT-Konzeption keinerlei Marktchancen mehr.

Ende 1988 wurde aus dem SGMP-Vorschlag das SNMP-Protokoll. Ein knappes weiteres Jahr ging ins Land und man erklärte SNMP auf einem zweiten IAB-Meeting zum »De-facto-Standard«. Im Mai 1990 erhielt es den Status *recommended*, der dann zu einem *Full IAB Standard* wurde.

Die unterschiedlichen Konzeptionen beider Protokollphilosophien lassen sich folgendermaßen bewerten:

Die Gemeinsamkeiten der Protokolle liegen zum einen in der Verwendung der Darstellungssprache ASN.1 (*Abstract Syntax Notation*) für die Beschreibung und Definition der Managementinformationen SMI (*Structure and Identification of Management Information*) gemäß RFC 1155 sowie zum anderen in der Objekterverwaltung in MIBs (*Management Information Base*) gemäß RFC 1156.

Im Unterschied zu SNMP arbeitet CMIP verbindungsorientiert und impliziert dadurch einen höheren Overhead. SNMP benutzt UDP als verbindungsloses Protokoll, wodurch der Informationsfluss wesentlich schneller erfolgt. Einem deutlichen Mehr an Funktionsvielfalt und umfangreicheren Objektdefinitionen im CMIP stehen bei SNMP eine recht einfache Implementierung und geringerer Ressourcenverbrauch gegenüber. Für die Kommunikation zwischen *Manager* und *Agent* verwendet das SNMP das *trap directed polling*, das heißt, der Hauptanteil an Aktivitäten wird auf die Managerkomponente verlagert. Das *Event-based-Modell* von CMIP reduziert diesen Aufwand für den Manager und überträgt ihn größtenteils auf die Agenten. Der Einsatz von SNMP sollte auf lokale Netzwerke (LAN) begrenzt werden. Der voraussichtlich relativ hohe Polling-Traffic würde ein Weitverkehrsnetz (WAN) über Gebühr belasten; hier ist eindeutig CMIP zu favorisieren.

An den Tatsachen kommt man jedoch trotz potenzieller technologischer Vorteile des CMIP nicht vorbei. Der aktuelle Markt wird primär von SNMP-Produkten beherrscht. Einfachheit und Kostenvorteile sind offenbar die dominanten Einsatzkriterien. Die technologisch fällige SNMP-Weiterentwicklung zu SNMPv2 wurde allerdings derart komplex, dass eine Akzeptanz der Hersteller ausblieb. Die viel diskutierte Version 3 sollte als Synthese von SNMPv1 und SNMPv2 weite Verbreitung finden.

SNMP-Architektur

SNMP als Protokoll im herkömmlichen Sinn kann grundsätzlich nicht problemlos in das ISO/OSI-Schichtenmodell integriert werden, da es sich wie eine »Pipeline« durch den gesamten Protokollstack zieht und den Transport von Informationen übernimmt, die innerhalb einer MIB (*Management Information Base*) für jedes individuelle Objekt einer Protokollschicht dargestellt werden können. [Abbildung 6-16](#) zeigt dies beispielhaft.

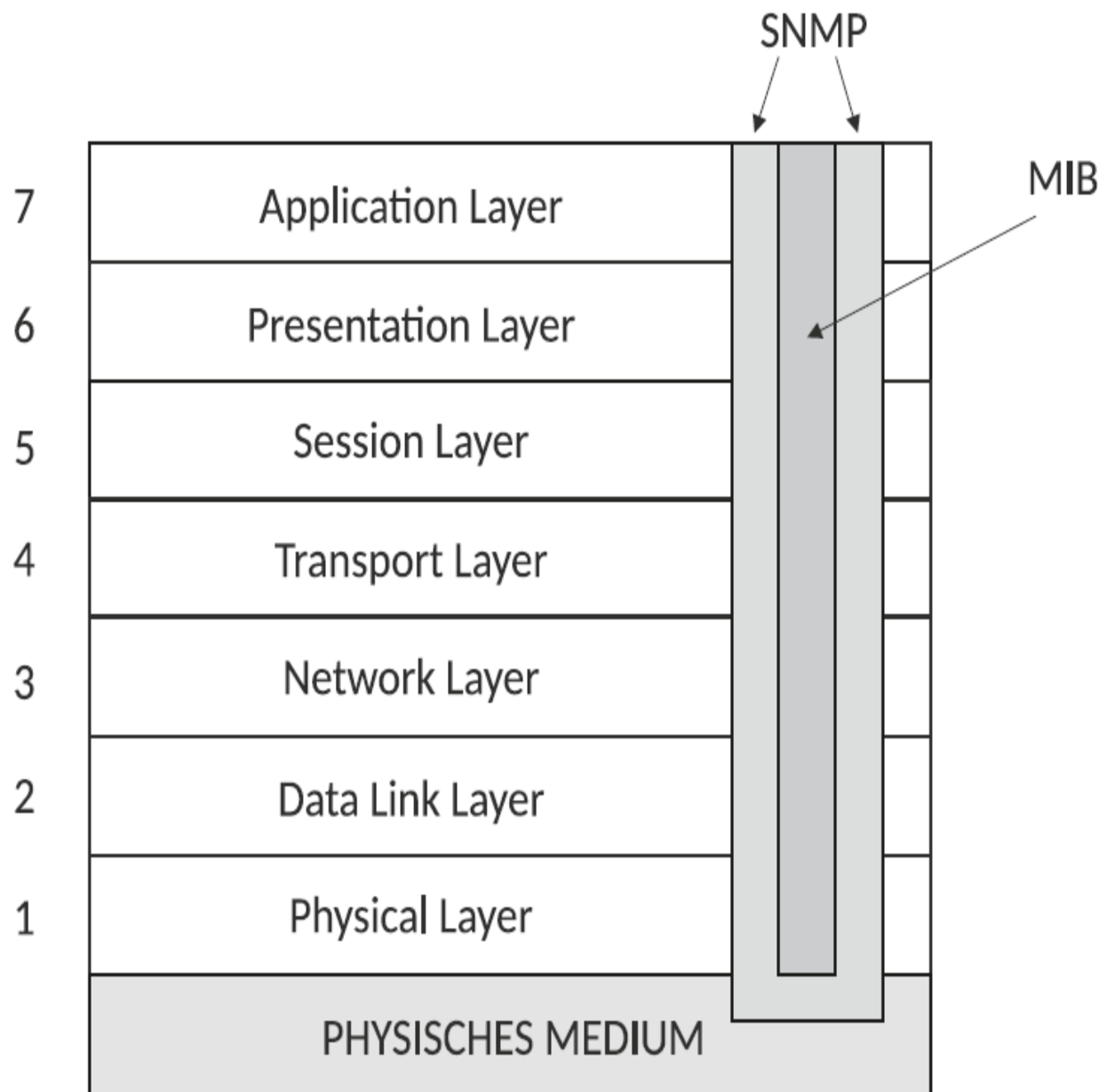


Abb. 6–16 SNMP-Architektur

HINWEIS

Mit der besonderen Philosophie von SNMP lassen sich sowohl Adressinformationen des physischen Layers (z.B. Adresse des

Netzwerkcontrollers) als auch *Timer* aus der TCP-Schicht ermitteln.

SNMP-Komponenten

Die in Hard- und Software abgebildeten Komponenten eines SNMP-Managementsystems lassen sich in drei Kategorien unterteilen:

- Network Management Stations (NMS)
- SNMP-Agenten
- Proxy-Agenten

In einem Netzwerk mit zahlreichen Netzkomponenten gibt es eine zentrale Managementinstanz, die für die Verwaltung der Netzwerkkomponenten zuständig ist. In der Regel handelt es sich dabei um einen leistungsfähigen Rechner, der als Plattform über Managementanwendungen (NMA = *Network Management Application*) verfügt, die ein *Monitoring* und *Operating* der Netzwerkelemente (Hosts, Gateways, Router, Switches, Terminal-Server, XTerminals, Desktops usw.) vornehmen. Dabei werden in den einzelnen Netzwerkelementen spezielle *Management Agents* (SNMP-Agenten) eingerichtet, die auf Basis einer SNMP-Verbindung mit der NMA die Durchführung von Managementaufgaben ermöglichen.

Sollen auch Geräte oder Elemente (NE = Netzwerkelemente) verwaltet werden, die nicht Bestandteil des IP-Netzwerks und damit auch nicht über SNMP erreichbar sind, so werden sogenannte *Proxy Agents* benötigt. Dieser besondere Agententyp wird einerseits mit der Fähigkeit ausgestattet, SNMP als Protokoll zuzulassen (um damit die Erreichbarkeit durch die NMA zu gewährleisten), und andererseits kommuniziert dieser mit den Rechnern eines protokollfremden Netzwerks über das dort eingesetzte Protokoll. Es ist somit möglich, ein Netzwerkobjekt, das keinen Zugang zu einem TCP/IP-Netzwerk besitzt, über solche Proxy-Agenten zu integrieren. Allerdings ist es erforderlich, auf diesen Agenten einen Multi-Protokollstack (*Gateway*) einzurichten, damit zu beiden Netzwerken eine operative Verbindung etabliert werden kann (siehe [Abb. 6-17](#)).

Network Management Station (NMS)

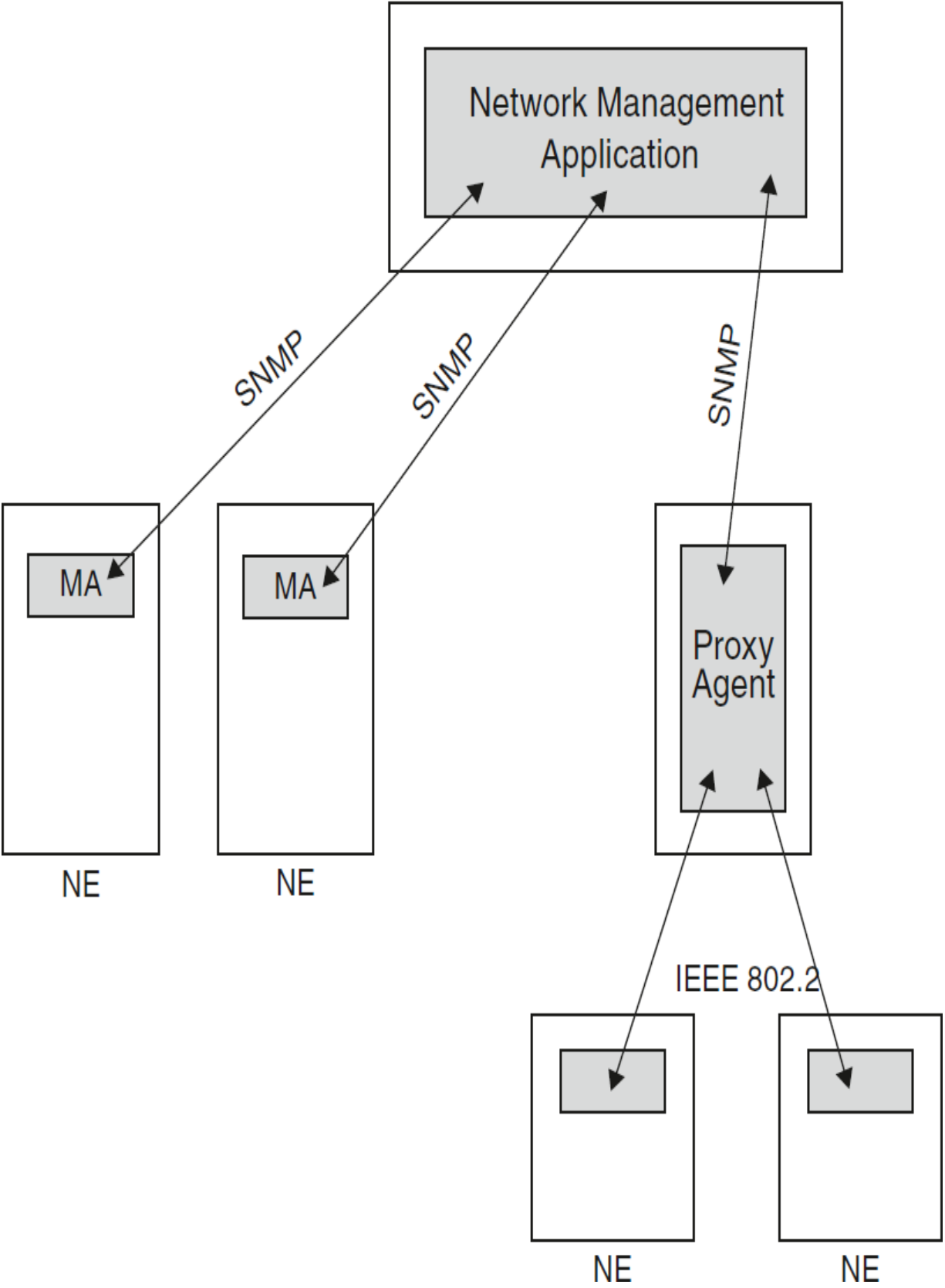


Abb. 6–17 SNMP-Komponenten

Für die Darstellung und den Transport der managementrelevanten Daten werden drei Objektkategorien verwendet, die nachfolgend näher beschrieben sind:

- *SMI (Structure and Identification of Management Information)*
Modell zur Beschreibung der Struktur von benötigten Informationen
- *MIB (Management Information Base)*
Vereinbarte Standards zur Beschreibung der individuellen Netzwerkobjekte und ihrer Subobjekte. Dem Begriff liegt keine reale Datenbank zugrunde, sondern lediglich das Verfahren und die Beschreibung.
- *SNMP (Simple Network Management Protocol)*
Transportprotokoll zur Realisierung der Kommunikation zwischen der *Network Management Station* (NMS) und den *Management Agents* (MA)

Structure and Identification of Management Information (SMI)

Die SMI stellt das administrative Umfeld zur Definition von Managementobjekten sowie des Protokolls dar, das auf diese Objekte zugreifen soll. Die hierbei verwendete

Beschreibungssprache ist die *Abstract Syntax Notation One* (ASN.1). Es folgt eine Übersicht der fünf Beschreibungsfelder:

- *OBJEKT*

Der *object descriptor* wird zur Bestimmung des Objekttyps (z.B. *sysDescr*) verwendet, der *object identifier* für den Namen des Objekts selbst (z.B. *system 1*).

- *SYNTAX*

Gemäß RFC 1155 wird eine abstrakte Syntax für das *Managed Object* (MO) eingeführt. Diese kann als *SimpleSyntax* aus Integer, Octet String, Object Identifier oder Null bestehen oder aber auch als *ApplicationSyntax* eine Netzwerkadresse, einen Zähler oder auch Zeittakte verwenden.

- *DEFINITION*

Die *definition* beinhaltet eine textuelle Beschreibung des Objekts in freier Formulierung.

- *ACCESS*

Durch dieses Feld wird festgelegt, wie der Zugriff auf das Objekt zu erfolgen hat: nur lesend (*read-only*), lesend und schreibend (*read-write*), nur schreibend (*write-only*) oder kein Zugriff (*not-accessible*).

- *STATUS*

Die drei potenziellen Statusattribute lauten: *mandatory*, *optional* oder *obsolete* (verbindlich vorgeschrieben, wahlweise oder veraltet bzw. überholt).

Nachfolgend ist beispielhaft eine entsprechende Objektbeschreibung (*Managed Object*) anhand der ASN.1 dargestellt:

```
OBJECT

sysDescr { system 1 }

SYNTAX      Octet String

[ This value should include the full name and version identificat

ACCESS      read-only.

STATUS      mandatory.
```

Eine derartige Beschreibung des *MO* allein reicht jedoch nicht aus. Es fehlt die genaue Identifizierung bzw. eine genaue »Wegbeschreibung«, um das Objekt zu erreichen. Zu diesem Zweck wird der sogenannte *ASN.1 Object Identifier* verwendet. So wird für dieses Objekt der Zugriffspfad 1.3.6.1.2 für die

Management Information Base (MIB) verwendet. Durch die Erweiterung des Pfads auf 1.3.6.1.2.1.1 ist der Objekttyp *sysDescr* mit dem Identifier *system 1* eindeutig identifiziert (Standard-MIB). Die Zuordnung von *Managed Object* und *object identifier* geht aus [Abbildung 6–18](#) hervor:

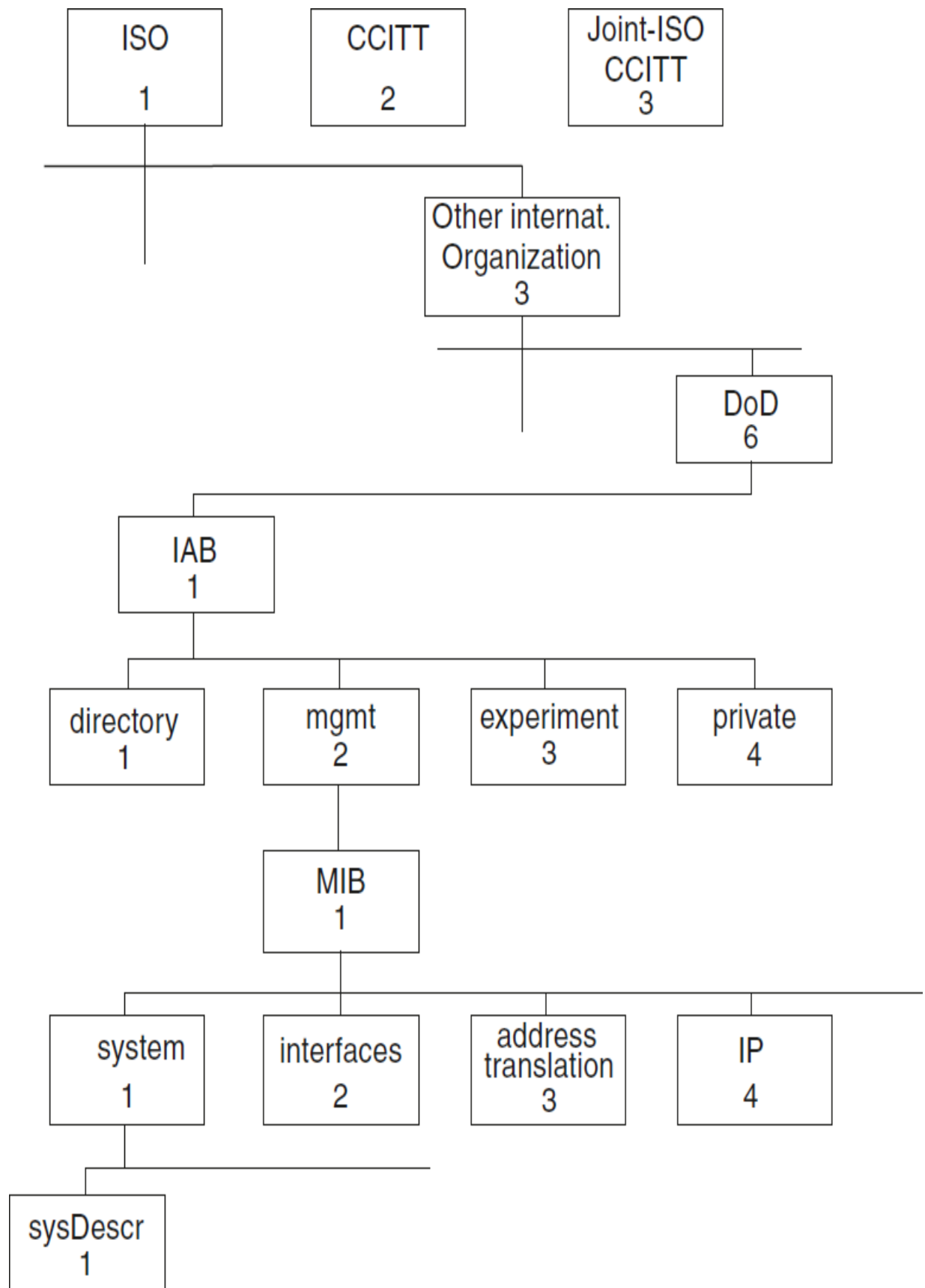


Abb. 6–18 SMI-Object-Identifizier 1.3.6.1.2.1.1.1

Management Information Base (MIB)

Unter der *Management Information Base* (MIB) versteht man eine virtuelle Datenbank, die konkrete Beschreibungen für Objekte (*Managed Objects*) enthält, die dem SNMP-Agenten bei seiner Kommunikation mit der *Network Management Station* (NMS) zur Verfügung stehen. Die MIB stellt Strukturen (*Pointer*) zur Verfügung, die einen direkten Zugriff auf Einzelinformationen der Objekte erlauben.

Man unterscheidet konzeptionell Informationen aus der Standard-MIB und nicht standardisierte Objekte, die jedoch SMI-konform sein müssen. Die Standard-MIB entwickelte sich zunächst aus der MIB-I, die ausführlich im RFC 1156 beschrieben wird. Sie wurde mittlerweile jedoch von der erweiterten MIB-II abgelöst.

Die einzelnen Objekte der MIB sind in Gruppen angeordnet, die sich einerseits auf Informationen des Gesamtsystems (bzw. Rechners) und andererseits auf einzelne Netzwerkprotokolle (in unterschiedlichen Netzwerkschichten) beziehen. In den nachfolgenden Tabellen sind die Objekte der einzelnen MIB-Varianten dargestellt:

MIB-I

Gruppe	Beschreibung	Anzahl Objekte
System	Informationen über das eigene System	3
Interfaces	Informationen über das (die) Interface(s)	22
At	Informationen über das Adress-Mapping	3
Ip	Internet Protocol (Zähler, Adressen, Routing usw.)	33
Icmp	ICMP-Informationen	26
Tcp	TCP-Informationen	17
Udp	UDP-Informationen	4
Egp	EGP-Informationen	6

HINWEIS

Ein SNMP-Agent braucht lediglich die für ihn relevanten Gruppen zu implementieren. Für einen einfachen Host ist es daher beispielsweise nicht erforderlich, die Gruppe *egp* zu verwenden. Eine teilweise Verwendung von Objekten innerhalb einer Gruppe ist jedoch nicht möglich.

MIB-II

Gruppe	Beschreibung	Anzahl Objekte
System	Informationen über das eigene System	7
Interfaces	Informationen über das (die) Interface(s)	23
At	Informationen über das Adress-Mapping	3
Ip	Internet Protocol (Zähler, Adressen, Routing usw.)	38
Icmp	ICMP-Informationen	26
Tcp	TCP-Informationen	19
Udp	UDP-Informationen	7
Egp	EGP-Informationen	18
transmission	reserviert für medienspezifische Informationen aus dem - Experimentalzweig (vgl. Abb. 6-18 : object identifier 1.3.6.1.3)	0
Snmp	Informationen über SNMP-Netzwerkdaten	30

Nachstehend einige Objekt-Beispiele der einzelnen Gruppen gemäß Spezifikation des RFC 1213 der MIB-II:

- system (1)

- sysDescr* vollständige Systembeschreibung
- sysObjectID* Objekt-Identifikation des Herstellers
- sysUpTime* Zeit seit dem letzten Systemstart
- sysContact* Name der Kontaktperson dieses Systems
- sysServices* von diesem System angebotene Dienste

- interfaces (2)

- ifIndex* Interface-Nummer
- ifDescr* Beschreibung des Interface
- ifType* Beschreibung des Interface-Typs
- ifMTU* Größe der MTU (Maximum Transfer Unit)
- ifAdminStatus* Statusinformation des Interface
- ifLastChange* Zeitpunkt der letzten Statusänderung
- ifINErrors* Anzahl Inbound-Pakete mit Fehlern
- ifOutDiscards* Anzahl fehlerhafter/verworfenener Outbound-Pakete

- at (3)

- ip (4)

- icmp (5)

- tcp (6)

- udp (7)
- egp (8)
- snmp (11)

Weitere Gruppen der MIB-II sind mittlerweile ergänzt worden bzw. sind in Vorbereitung:

- MIB-II-Gruppe 10:Transmission

z.B. Ethernet RFC 1643

- MIB-II-Gruppe ifExtensionsRFC 2233

12:

- MIB-II-Gruppe appletalkRFC 1742

13:

- MIB-II-Gruppe OSPFRFC 1850

14:

- MIB-II-Gruppe BGPRFC 1269

15:

- MIB-II-Gruppe RMON RFC 1757

16:

- MIB-II-Gruppe dot1dBridge (Bridges)RFC 1493

17:

- MIB-II-Gruppe phiv (DecNet Phase IV)RFC 1559

18:

- MIB-II-Gruppe RIPv2RFC 1724

23:

- usw.

Private-MIB

Die Vielzahl von Herstellern und die Kurzlebigkeit zahlreicher Produkte erlauben es nicht, die relevanten Informationen in der Standard-MIB aktuell zu halten. Für diesen Zweck ist es möglich, sogenannte *Private-MIBs* zu implementieren. Mit dem *object-id-Prefix* 1.3.6.1.4 lässt sich eine beliebige Zahl herstellerspezifischer Objekte beschreiben, die für die NMAs (*Network Management Application*) zugänglich gemacht werden können. Die Konventionen, die für MIB-II-Objekte gelten, müssen allerdings auch hier eingehalten werden (ASN.1, SMI usw.). Sobald neue Produkte auf den Markt gebracht werden, wird in den meisten Fällen die spezifische MIB vom Hersteller bereits beigelegt.

MIB-Adressierung

Bislang wurden bei der Erörterung von Objekten lediglich die durch Zuweisung von Object Identifiers generierten *Knotenelemente* behandelt. Knotenelemente sind innerhalb einer MIB jedoch selbst nicht adressierbar, d.h., man kann sie weder abfragen noch modifizieren und auf einen anderen Wert setzen. Dazu benötigt man die sogenannten Blattelemente (*leaves*). Sie sind durch die vom Knotenelement ausgehende Beschreibung eines Managed Object gemäß ASN.1-Konventionen herleitbar.

Man unterscheidet zwei verschiedene Typen dieser Blattelemente: Ein eindimensionales Element besitzt keine weitere Teilmenge von Elementen und kann somit direkt adressiert werden. In dem Fall wird das Suffix »0« verwendet. Die Adressierung von *sysDescr* aus der MIB-II-Systemgruppe wird somit mit *sysDescr.0* bzw. mit 1.3.6.1.2.1.1.1.0 vorgenommen. Ein zweidimensionales Element wird durch eine mehrspaltige Tabelle dargestellt und entsprechend adressiert. Hier muss, entsprechend der Position in der Spalte, zur Identifikation ein Suffix größer »0« verwendet werden.

Mit dem Object Identifier 1.3.6.1.2.1.2.2.1 wird das Objekt *ifEntry* innerhalb der Interfaces-Gruppe angesprochen. Es

handelt sich bei diesem Objekt allerdings um ein Tabellen-Objekt, bestehend aus 3 Zeilen und 22 Spalten. Zur eindeutigen Adressierung wird nun der obige Object Identifier um die jeweilige Spaltennummer und die Zeilennummer erweitert.

Zu dem Objekt *ifEntry* gibt es genau 22 weitere Objekte, die zusätzliche Informationen liefern. In diesem Fall gibt jedes der 22 Objekte Informationen für jedes vorhandene Netzwerk-Interface des über SNMP kontaktierten Rechners an. Insgesamt sind drei Interfaces eingebaut. Die Abfrage des *ifEntry*-Objekts ergibt folgendes Ergebnis:

Das Präfix lautet:

```
1.3.6.1.2.1.2.2.1.2.2.1 bzw. iso.org.dod.iab.m
```

```
2.interfaces.ifTable.ifEntry
```

```
.ifIndex.11.1 1
```

```
.ifIndex.21.2 2
```

```
.ifIndex.31.3 3
```

.ifDescr.12.1 lo0

.ifDescr.22.2 en0

.ifDescr.32.3 tr0

.ifType.1 3.1 softwareLoopback

.ifType.2 3.2 ethernet-csmacd

. . .

...

.ifSpecific.122.1

.ifSpecific.222.2

.ifSpecific.322.3

SNMP-Anweisungen

Das Instrumentarium zur Informationsbeschaffung sind lediglich einige wenige *SNMP-Messages*, die in Form nachstehend aufgeführter SNMP-Befehle in den einzelnen SNMP-Agenten unterschiedlicher Betriebssysteme (Windows, UNIX, Linux, OS/2, MS-DOS) implementiert sind:

- *Get-Request*

Die NMS führt eine gezielte Abfrage der MIB-Variablen von Knotenelementen beim entfernten NMA durch. Beispiel: *snmpget 2.4.0.9 1.3.6.1.2.1.1.1.0* liefert den Wert der *sysDescr*-Variablen.

- *GetNext-Request*

Die NMS führt eine Abfrage der MIB-Variablen von Blattelementen beim entfernten NMA durch. Die Anzahl der Elemente einer Tabelle ist zumeist nicht bekannt. Beispiel: *snmpnext 2.4.0.9 1.3.6.1.2.1.2.2.1.2.2.1* liefert die Ausgabe obiger Tabelle.

- *Get-Response*

Antwort auf den Get-Request und GetNext-Request durch den NMA

- *Set-Request*

Die NMS kann den Wert einiger MIB-Variablen beim entfernten Rechner verändern. Beispiel: *snmpset 2.4.0.9 1.3.6.1.2.1.1.4.0 Karl Meier* speichert die Zeichenkette *Karl Meier* in die MIB-Variable *sysContact.0*

- *Trap*

Abhängig von definierten Ereignissen kann der NMA einen Trap an die NMS senden. Es wird dazu der *UDP well known port 162* verwendet. Andere *SNMP-Messages* benutzen Portnummer 161, um eine unabhängige Verarbeitung von Traps und der Get-/Set-Kommandos zu ermöglichen.

SNMP-Message-Format

SNMP-Messages werden gemäß OSI-Definition im PDU-Format (*Protocol Data Units*) abgebildet. Es lassen sich grundsätzlich zwei SNMP-PDU-Typen unterscheiden: Die normale PDU umfasst die *Get-Request*-, *GetNext-Request*- und die *Set-Request-Messages* einschließlich ihrer Responses (siehe [Abb. 6–19](#)). Das Trap-PDU-Format sieht ein wenig anders aus.

Request/Response-PDU-Format

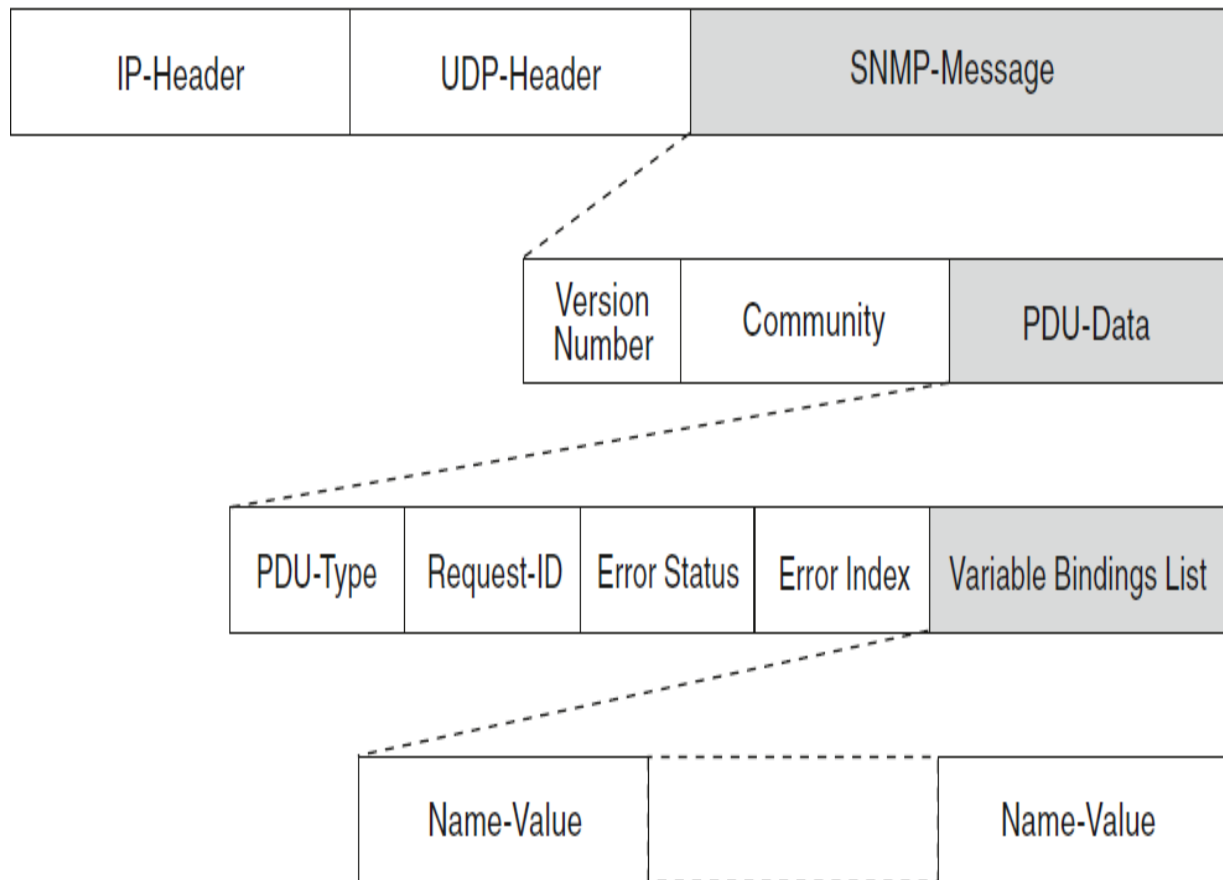


Abb. 6–19 Request/Response-PDU-Format

- *Version Number*

Der Wert »0« repräsentiert die Versionsnummer 1, Wert »1« bedeutet Version 2 usw.

- *Community*

Der 5 Bytes umfassende *Community Name* stellt eine Art SNMP-Authentifizierung zum Abfragen und Verändern der Werte von MIB-Variablen dar. Nur Kommunikationspartner mit identischem *Community Name* können am operativen SNMP teilnehmen.

- *PDU-Type*

Angabe des SNMP-Message-Typs bzw. des verwendeten SNMP-Befehls (Get-Request, GetNext-Request usw.)

- *Request-ID*

Eindeutige Identifizierung des Requests

- *Error Status*

Folgende Statuswerte sind möglich:

- 0kein Fehler
- 1PDU zu lang
- 2Name existiert nicht, Wert nicht vorhanden
- 3falscher Wert
- 4Read-only-Feld, Modifikation nicht möglich
- 5allgemeiner Fehler

- *Error Index*

Zusatzinformationen zum Status des Fehlers

- *Variable Bindings List*

Eine Liste variabler Größe mit *Object Identifier* und seinem Wert ist hier hinterlegt. Bei einem *Request* sind die Wertefelder ausgenullt. Die Response enthält hier das Ergebnis einer Abfrage.

SNMP-Sicherheit

Zur Gewährleistung einer minimalen Sicherheit im SNMP-Datenverkehr gibt es den *Community-String*, eine Zeichenkette,

die in jeder *SNMP-Message* übertragen wird. Seine Auswertung erfolgt seitens des SNMP-Agenten. Ferner kann der zuständige Administrator dafür sorgen, dass infolge einer eingeschränkten Sicht auf die MIB-Baumstruktur nur ein kleiner Teil der MIB-Variablen zugänglich ist. Darüber hinaus lässt sich eine selbst festgelegte Zugriffsmethode (*read-only*, *read-write*, *write-only* oder *none*) definieren und einer *Community* zuordnen. Die Default-Einstellung für die Standard-Community *public* lautet *read-only*; alle weiteren Einstellungen können individuell gestaltet werden.

Die erforderlichen Definitionen werden (bei UNIX-Systemen) in der Datei */etc/snmpd.conf* des SNMP-Agenten vorgenommen.

HINWEIS

Bei näherer Betrachtung sollte man sich immer vor Augen führen, dass ein derartiges Sicherheitskonzept äußerst fragwürdig ist. Was nützt ein sicherheitsrelevanter *Community-String*, wenn er unverschlüsselt über das Netzwerk übermittelt wird und demzufolge problemlos abgehört werden kann? Die daraus resultierenden Unwägbarkeiten haben dazu geführt, dass zahlreiche Hersteller in ihrer privaten MIB die Möglichkeit einer Modifizierung von Werten bestimmter MIB-

Variablen nicht vorsehen und grundsätzlich das Zugriffsrecht *read-only* verwenden.

SNMP-Nachfolger

Aus verschiedenen, vorhergehend teilweise aufgeführten Gründen ergab sich zwangsläufig die Weiterentwicklung des SNMP-Protokolls. So sollte mit den neuen Versionen insbesondere den bisherigen Mängeln in der Sicherheitsvorsorge Rechnung getragen werden.

SNMPv2

Bei SNMPv2 stehen verschiedene Verschlüsselungsverfahren zur Verfügung, um die Sicherheit der SNMP-Kommunikation zu erhöhen. Die konventionellen Verschlüsselungsverfahren beruhen auf einem geheimen Schlüssel, der vom Sender und vom Empfänger gemeinsam benutzt wird. Die Verschlüsselung erfolgt auf der Seite des Senders, und nach Übertragung der verschlüsselten Daten werden diese empfangsseitig mithilfe des geheimen Schlüssels entschlüsselt.

- *DES (Data Encryption System)*

Bei DES besteht das Verschlüsselungsprinzip in einer 16-fachen Wiederholung eines Mischungsprozesses, der nach jedem Schritt eine Unterteilung der 64-Bit-Datenfolge in eine

linke und eine rechte Hälfte vollzieht. Mit einem sogenannten *F-Modul* werden beide Hälften modifiziert und verglichen. Übereinstimmungen werden mit einer »0«, keine Übereinstimmungen mit einer »1« gekennzeichnet. Die so entstehende neue Folge wird zur neuen rechten Hälfte der Ergebnisfolge. Bei jeder der 16 Wiederholungen wird ein neuer Schlüssel verwendet. Durch weiteres Zerlegen und Zusammenführen wird die Spur des ursprünglichen Schlüssels derart verwischt, dass es unmöglich ist, diesen jemals zu rekonstruieren.

- *PKI*

Public-Key-Verschlüsselungsverfahren (*PKI = Public Key Infrastructure*) verwenden einen öffentlichen Schlüssel für die Datenverschlüsselung und einen geheimen Schlüssel für die Entschlüsselung der Daten. Dieser Schlüssel lässt sich jedoch nicht aus dem öffentlichen Schlüssel und/oder dem Verschlüsselungsalgorithmus ermitteln.

- *RSA*

RSA (benannt nach den drei maßgeblich beteiligten Mathematikern Ronald L. Rivest, Adi Shamir, Leonard Adleman) ist ein Verschlüsselungsprinzip, bei dem jeder Netzwerkteilnehmer je zwei Primzahlen und einen zu dem Produkt der um 1 verminderten Primzahlen teilerfremden Verschlüsselungsexponenten wählt. Teilnehmer A wählt die

Primzahlen p und q , der Verschlüsselungsexponent lautet e . Es wird lediglich das Produkt $n = p \times q$ sowie der Exponent e veröffentlicht. Der Verschlüsselungsalgorithmus lautet dann:

$V(m) = me(mod\ n)$. Die Variable m steht hier für die Information, die verschlüsselt werden soll. Der Entschlüsselungsalgorithmus lautet für den verschlüsselten Text y : $E(y) = yd(mod\ n)$. Die Zahl d kann nur berechnet werden, wenn p und q bekannt sind. Betrachtet man allein die Sicherheitsfrage bei SNMP und SNMPv2, so scheint der Problematik einer mangelhaften Ausprägung in der SNMPv1 mit einem übertrieben umfangreichen Sicherheitskonzept in der Nachfolgeversion nicht effektiv begegnet worden zu sein. Denn die Verwaltung von Verschlüsselungskonzepten ist sehr aufwendig und bedarf einer genauen Abstimmung, wenn Schlüssel gewechselt werden müssen. Ein weiteres Problem ist die zurzeit noch ungeklärte Situation des Einsatzes von Verschlüsselungsverfahren, wie z.B. des DES, außerhalb der USA. Hier besteht noch ein Exportverbot für bestimmte Schlüssellängen.

Weitere Modifikationen im SNMPv2 als Überblick:

- Bei der Abfrage von Tabellen aus MIB-Bereichen wird der bislang eingesetzte *GetNext-Request* durch den neuen *GetBulk-Request* ersetzt. Eine deutliche Reduzierung der

Netzbelastung und eine spürbare Steigerung der Auslesegeschwindigkeit sind die Folge.

- Neue Verfahren wurden entwickelt, um die Wartung (Einfügen/Löschen) von Tabellen- bzw. Spaltensegmenten innerhalb einer MIB zu verbessern.
- Es wurde eine Vereinheitlichung des Aufbaus aller SNMP-PDUs realisiert.
- Neue Datentypen und Objektmakros sind entwickelt worden.
- Die »Sub Agent«-Konzeption ermöglicht die dynamische Erweiterung von mehreren *Sub Agents* unter einem einzigen *MasterAgent*.
- Einführungen von *Party-MIBs*. Informationen zur Autorisierungssteuerung sind hier hinterlegt. Die Party-MIB stellt eine deutlich sicherheitsorientierte Weiterentwicklung des Community-Prinzips in SNMPv1 dar. Ein hoher Aufwand für die erstmalige Konfiguration des NMS und aller Agenten ist allerdings zu berücksichtigen.
- Ein direkter Informationsaustausch mehrerer NMS untereinander kann durch einen *InformRequest* vorgenommen werden. Dadurch werden dezentrales Management und eine Lizenzkostenreduzierung bei weltweitem Einsatz durch internationale Konzerne möglich (Beispiel: Übernahme des Netzwerkmanagements der europäischen Filialen durch die amerikanischen

Zweigstellen während der Nachtphase in Europa und umgekehrt – Übernahme der Softwarelizenz und ihre Ausnutzung über die gesamten 24 Stunden eines Tages).

- Ein Parallelbetrieb von SNMPv1 und SNMPv2 ist möglich, da eine weitgehende Kompatibilität beider Protokolle untereinander besteht.

HINWEIS

Die Komponentenarchitektur (siehe [Abb. 6–20](#)) des SNMPv2 hat sich gegenüber der Vorgängerversion nur wenig verändert. Die *Manager-to-Manager-Funktionalität* sowie die neue *Sub-Agent-Konzeption* stellen die wesentlichsten Unterschiede dar.

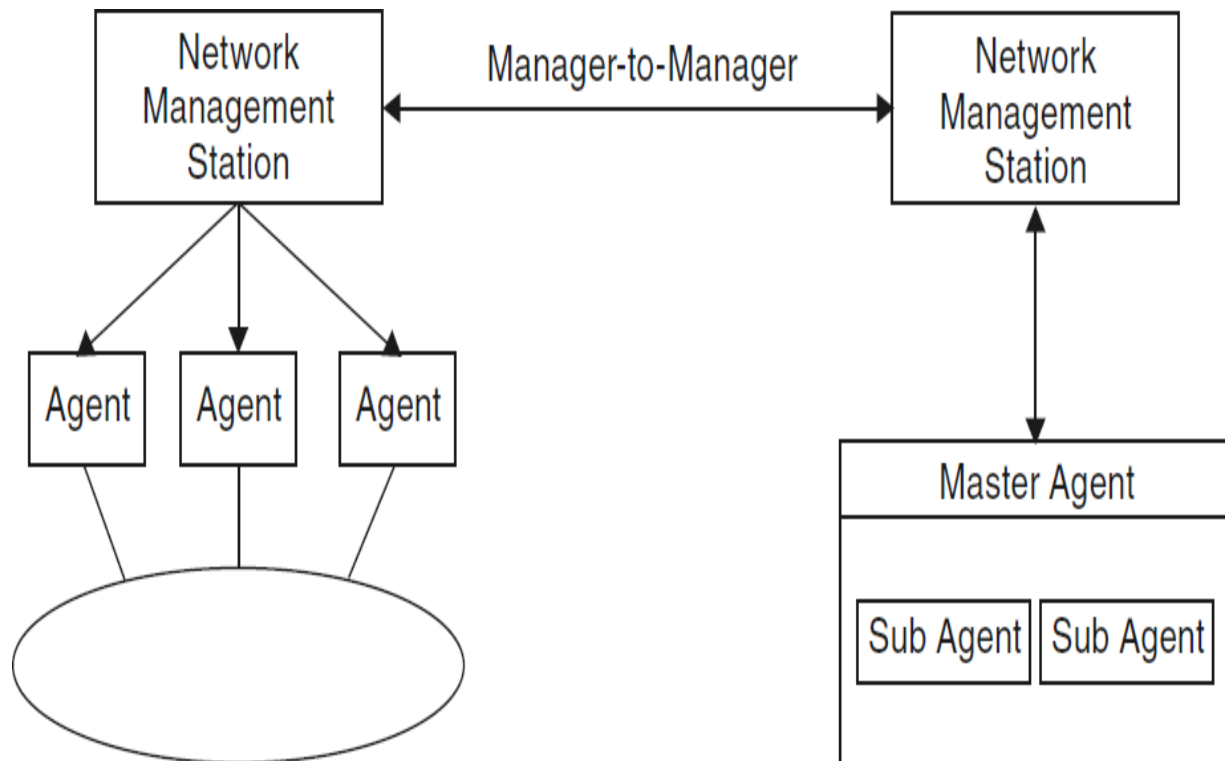


Abb. 6–20 SNMPv2-Komponenten

SNMPv3

Die Weiterentwicklung des SNMP in der Version SNMPv2 erreichte teilweise ein derart komplexes Ausmaß, dass eine Akzeptanz der Hersteller aktiver Netzwerkkomponenten ausblieb. Dies wurde unter anderem auch dadurch begünstigt, dass die ursprünglich geplanten SNMP-Spezifikationen insbesondere hinsichtlich Authentifizierung und Zugriffssicherheit nicht umgesetzt wurden und damit wesentliche Bestandteile eines zeitgemäßen Managementprotokolls fehlten. Anfang 1999 wurden daher erste konkrete Entwürfe (RFC 2570 vom April 1999: *Introduction*

to Version 3 of the Internet-standard Network Management Framework) zu einer neuen Protokollarchitektur, dem SNMPv3, veröffentlicht.

Einige wichtige Ziele wurden bereits zu Beginn formuliert:

- Basis für die Weiterentwicklung sind SNMPv1 und SNMPv2
- Auf die Kompatibilität zu allen Vorgängerversionen ist zu achten.
- Authentifizierung und vertraulicher Datentransport müssen für den Datentransport über die gesamte »Managementstrecke« zwischen SNMP-Agent und dem NMA gewährleistet sein.
- Der nicht autorisierte Zugriff Dritter ist unbedingt zu verhindern. Dies soll über individuelle und konfigurierbare *views* erreicht werden.
- Die Konfiguration des SNMP-Managements muss über »ferngesteuerte« Prozeduren möglich sein (*remote configuration*).

Den bereits in SNMPv2 verfügbaren Spezifikationen einer Datendefinitionssprache, der MIB-Module und eines Transportprotokolls wurden die Kriterien Sicherheit und Administration hinzugefügt. Gegenüber den Vorgängerversionen wurde die SNMP-Architektur um die

sogenannte *SNMP-Entity* ergänzt. Dabei handelt es sich um eine SNMP-Engine und eine bzw. mehrere mit diesen verbundenen Anwendungen. Diese SNMP-Engine ist für die gesamte Managementfunktionalität verantwortlich und besteht aus folgenden Komponenten:

- *Dispatcher*
Steuerung und Verteilung der Meldungen
- *Message Processing*
Datenaufbereitung der verschiedenen Meldungstypen
- *Security*
Mechanismen zur Authentifizierung und Datensicherheit
- *Access Control*
MIB-Zugriffsschutz durch Formulierung individueller Regeln

Die SNMP-Engine, basierend auf dem im RFC 2574 vom April 1999 beschriebenen *User-based Security Model* (USB), stellt eine Anzahl verschiedener Dienste bereit, die als Basis für integrierte Applikationen (siehe RFC 2573 vom April 1999) dienen:

- *command generator*
Formuliert SNMP-Requests wie beispielsweise *snmpget*, *snmpgetnext* oder *snmpset* aus.
- *command responder*

Übernimmt die *SNMP-Requests* und führt diese aus.

- *notification originator*

Überprüft den Status der Ressource und versendet Alarmmeldungen, sofern ein bestimmtes Ereignis eingetreten ist.

- *notification receiver*

Empfängt Alarmmeldungen (*traps*) und sorgt für die entsprechende Generierung von *SNMP-Responses*.

- *proxy forwarder*

Leitet *SNMP-Messages* ggf. unter Verwendung geeigneter Übersetzungsmechanismen weiter.

HINWEIS

In den RFCs 3410 bis 3418 (Dezember 2002) wurde die SNMPv3 zum Standard erhoben. Aktualisierungen hierzu erfolgten bislang bis April 2016 im RFC 7860, der sich mit Sicherheitsaspekten beschäftigt. Einzelheiten hierzu sind in den angegebenen RFCs nachzulesen.

Sicherheit im IP-Netzwerk

Allein mit dem Thema dieses Kapitels könnte man ein ganzes Buch füllen. So hat sich die Sicherheit im TCP/IP-Umfeld im Laufe der Zeit zu einem der wichtigsten Bereiche entwickelt und dabei zu entsprechenden Diskussionen in nahezu allen Unternehmen geführt. Die Sensibilität hierfür hat, nicht zuletzt durch einige bekannt gewordene Datenpannen, deutlich zugenommen. Wurde vor einiger Zeit nach dem Grundsatz verfahren: »Sicherheit ist uns zu teuer und können wir uns daher nicht leisten«, hat sich mittlerweile eine Kehrtwende vollzogen. In Zeiten zunehmender E-Business-Aktivitäten machen sich die Unternehmen zusehends Sorgen um ihre schützenswerten Ressourcen und investieren in die Entwicklung von Sicherheitskonzepten und -systemen.

Zugegeben, die Art und Weise, wie die Problematik angegangen wird, lässt oft noch zu wünschen übrig: In vielen Fällen werden hohe Investitionen für die Anschaffung und den Betrieb von Sicherheitssystemen getätigt in der Erwartung, nun das »Sicherheitsproblem« in den Griff bekommen zu haben. Was man dabei allerdings nur allzu gern vergisst: Die Sicherheitssysteme müssen von höchst qualifizierten Experten betreut werden, d.h., der interne Aufwand wird meist nur unzureichend berücksichtigt. Dies ist umso problematischer, als

nun ein gefährliches Sicherheitsbewusstsein entstanden ist («Wir haben ja schließlich ein aufwendiges Firewall-System installiert»). Lässt man nämlich ein Sicherheitssystem »wartungsfrei« laufen, d.h. ohne oder mit sehr wenig Personal, dann

- weiß man nicht, wie es um den tatsächlichen Sicherheitsgrad der Unternehmensressourcen bestellt ist (es fehlt eine Bewertungsinstanz für die eingesetzten Sicherheitssysteme – oft wird diese Aufgabe aber auch von externen Unternehmen angeboten, die sich auf solche Sicherheitsüberprüfungen (*Security Checks*) spezialisiert haben),
- kann man nicht rechtzeitig reagieren, wenn sich durch eine bestimmte Symptomatik der Ereignisse in der Protokolldatei eines Sicherheitssystems ein *Cracker-Angriff* ankündigt (auch wenn der Angriff selbst vom implementierten Sicherheitssystem abgewehrt wird, so müssen gegebenenfalls bei weiteren Angriffen geeignete Gegenmaßnahmen eingeleitet werden),
- wird ein dennoch erfolgreicher Angriff auf die Unternehmensressourcen erst viel zu spät oder überhaupt nicht bemerkt und führt somit zu unkalkulierbaren Schäden (Datenvernichtung, Datenmanipulation),
- können die Sicherheitssysteme nicht kontinuierlich auf dem neuesten technologisch relevanten Stand gehalten werden

(dies ist sehr wichtig, da sich die Vorgehensweisen von Crackern häufig ändern und demzufolge die Sicherheitssoftware der neuen Situation angepasst werden muss),

- gelingt es häufig nicht, die eigene Unternehmensstrategie für den Bereich Datensicherheit wunschgemäß umzusetzen, da es niemanden gibt, der sich in der Thematik wirklich auskennt und somit zur Gestaltung einer entsprechenden Unternehmenspolitik beitragen kann (ein »Outsourcing« dieser Aufgaben hilft hier meist auch nicht weiter, da das Verständnis für unternehmenseigene Anforderungen extern oft nur unzureichend ausgeprägt ist).

HINWEIS

Das Thema Sicherheit im Netzwerk hat mittlerweile den entsprechenden Stellenwert innerhalb der Unternehmen erreicht und wird auch weiterhin spürbar an Bedeutung gewinnen.

Interne Sicherheit

Spricht man von Datensicherheit in Netzwerken, so bringt man heutzutage oft nur einen einzigen Aspekt mit diesem Thema in

Verbindung. Es handelt sich dabei um Probleme, die sich in folgenden Fragen äußern:

- Wie kann eine Anbindung an das weltweite verteilte Internet vorgenommen werden, ohne sich dabei den Gefahren eines Eindringens engagierter Hacker (bzw. krimineller Cracker) in das eigene Netzwerk bzw. den eigenen Rechner auszusetzen?
- Welche Sicherheitsmaßnahmen sind zu treffen und wie stark werden die anfallenden Kosten zu Buche schlagen?

Das Thema *Security* ist allerdings wesentlich komplexer und bezieht sich nicht nur auf die Gefahr externer Attacken auf das Netzwerk, sondern ebenso auf die interne Infrastruktur des Unternehmensnetzwerks. Es sind unter anderem Probleme wie der unerlaubte Zugriff auf sensible Daten aus den verschiedensten Unternehmensbereichen (Personal, Entwicklung, Finanzen usw.), die beabsichtigte oder unbeabsichtigte Zerstörung wichtiger Datenbestände oder die Infiltration durch Virenprogramme, die eine Inhouse-Security unbedingt erfordern.

Ein besonderes Problem in diesem Zusammenhang sind die zumeist in Klartext dem Netzwerk übergebenen Daten. Die Mehrheit der Netzwerkprotokolle, gerade im TCP/IP-Umfeld, kennt nur in sehr eingeschränktem Maße eine

Datenverschlüsselung. Mit entsprechenden Analyse-Tools können somit auch sensible Daten problemlos eingesehen und gegebenenfalls aufgezeichnet werden. In der bereits verfügbaren, aber noch sehr verhalten eingesetzten IP-Version 6 wird dem Problem der Datensicherheit auf Netzwerkebene in weit höherem Maße Rechnung getragen.

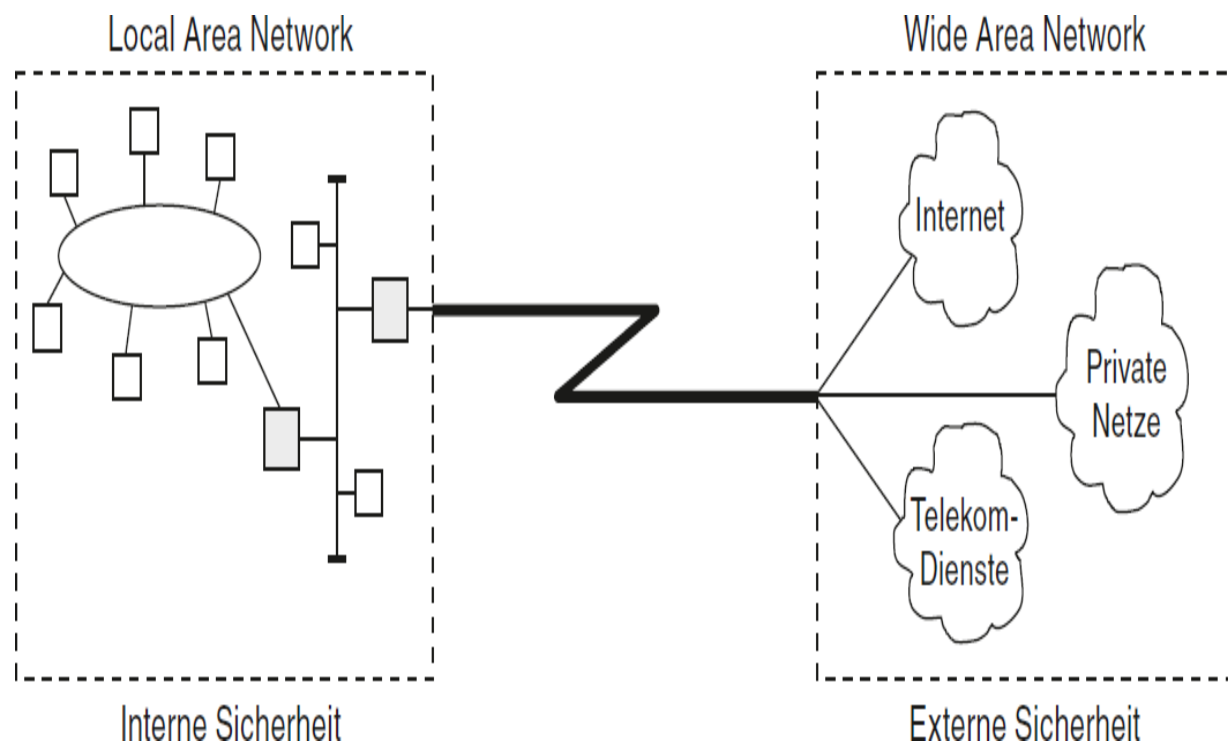


Abb. 7-1 Einteilung zwischen interner und externer Sicherheit

Bei allen geplanten Sicherheitsinstallationen sollte man jedoch zu keiner Zeit das richtige Augenmaß verlieren. Sicherheit hat natürlich auch ihren Preis (und damit ist nicht nur der monetäre Begriff gemeint). Sicherheit kann, sofern der Begriff äußerst restriktiv implementiert ist, eine Abteilung bzw. das

Unternehmen erheblich lähmen. Mit einer Sicherheitsphilosophie, die nur Selbstzweck ist, wird niemandem gedient sein. Es hat sicher wenig Sinn, einen Netzwerkverbund aus dem Finanzbereich nahezu völlig zu isolieren, denn er ist natürlich auf Informationen aus zahlreichen Bereichen des Unternehmens angewiesen. Der Netzwerk- und Rechnerzugriff muss daher so gestaltet sein, dass dem Informationsbedarf einerseits und der Security andererseits in einem ausgewogenen Verhältnis Rechnung getragen wird.

Einige Möglichkeiten, Maßnahmen bzw. Technologien zur Realisierung von Sicherheit umzusetzen, werden nachfolgend dargestellt. Dabei wird zwischen der internen und der externen Sicherheit unterschieden (siehe [Abb. 7-1](#)). Letztere beginnt genau dort, wo Daten das Unternehmen, d.h. das eigene Netzwerk, verlassen und vom LAN (*Local Area Network*) an ein WAN (*Wide Area Network*) übergeben werden.

Die interne Sicherheit umfasst zunächst Sicherheitskonzepte des jeweiligen Rechners samt seiner definierten Benutzer. Sie lässt sich darüber hinaus auf das unmittelbare LAN bzw. den LAN-Verbund des Unternehmens erweitern. In diesem Umfeld können verschiedene Maßnahmen zur Realisierung einzelner Sicherungsmechanismen getroffen werden.

Hardwaresicherheit

Ein gewisses Maß an Sicherheit kann bereits durch einen vernünftig organisierten Sicherungsschutz an der jeweiligen Hardware erreicht werden. Es ist beispielsweise darauf zu achten, dass sensible Systeme, die etwa für die Steuerung von Produktionsabläufen eingesetzt werden, nicht unmittelbar zugänglich sind. Die Unterbringung in eigens für sie vorgesehenen, abschließbaren Schranksystemen sollte obligatorisch sein. Außerdem sollten regelmäßig durchgeführte Wartungen den störungsfreien Betriebszyklus garantieren.

UNIX-Zugriffsrechte

Das Betriebssystem UNIX (respektive auch die Linux-Derivate) stellt selbst einige wichtige Sicherungskonzepte zur Verfügung, die zumeist auf die Tatsache zurückzuführen sind, dass es sich um ein Mehrbenutzersystem handelt, bei dem der konkurrierende Zugriff verschiedener Benutzer auf gleiche und verschiedene Datenbestände Schutzmechanismen unbedingt erfordert. Anhand der folgenden Beispiele sollen diese erläutert werden: Standard-UNIX-Rechte, SVTX (*Sticky Bit*), SGID (*Set Group Id*), SUID (*Set User Id*) und ACLs (*Access Control Lists*).

Standard-UNIX-Rechte

Nahezu sämtliche UNIX-basierenden Systeme (und dazu gehören auch die Linux-Derivate) ermöglichen für jedes Objekt, also jede Datei bzw. jedes Verzeichnis, die Zuordnung unterschiedlicher Zugriffsschutzmechanismen. Es existieren für jedes Objekt jeweils neun Zugriffsrechte, die folgendermaßen strukturiert sind:

Benutzerklasse:	Eigentümer	user
	Gruppe	group
	Andere	others
Rechte:	Lesen	read
	Schreiben	write
	Ausführen	execute

Auf jede einzelne der drei Benutzerklassen lässt sich ein Lese-, Schreib- oder auch Ausführungsrecht übertragen, wobei das Schreibrecht das Leserecht impliziert. Die Rechte werden mit den Kennbuchstaben *r* (*read*), *w* (*write*) und *x* (*execute*) versehen. Außerdem erhalten die neun verschiedenen Zugriffsrechte die in der folgenden Tabelle dargestellte oktale Wertigkeit:

Benutzerklasse	Leserecht (r)	Schreibrecht (w)	Ausführungsrecht (x)
user	400	200	100
group	40	20	10
others	4	2	1

Für eine Kombination von Zugriffsrechten werden die dargestellten Werte einfach addiert. Das Kommando, mit dem eine Bestimmung bzw. eine Festlegung der Zugriffsrechte für Dateien oder Verzeichnisse vorgenommen werden kann, lautet CHMOD, wobei folgende Syntax zu beachten ist:

```
chmod [Oktalwert] [Objektname]
```

Die aktuelle Anzeige der Zugriffsrechte eines Objekts (Datei oder Verzeichnis) kann mit dem List-Befehl (ls) und der Option *-l* zur Anzeige gebracht werden, so wie nachfolgend beispielhaft dargestellt:

```
-rwxr-xr-x 1 root bin      1316 May 5 1995 arch
```

-rwxr-xr-x	1	root	bin	299012	Apr	30	1995	bas
-rwxr-xr-x	1	root	bin	13316	Feb	14	1995	cat
-rwxr-xr-x	1	root	bin	12292	Apr	29	1995	chg
-rwxr-xr-x	1	root	bin	12292	Apr	29	1995	chm
-rwxr-xr-x	1	root	bin	12292	Apr	29	1995	cho
drwxr-xr-x	1	root	bin	20484	Apr	29	1995	dat
-rwxr-xr-x	1	root	bin	41988	Feb	14	1995	cpi
-rwxr-xr-x	1	root	bin	16388	Apr	29	1995	dd*
-rwxr-xr-x	1	root	bin	12292	Apr	29	1995	df*
-r-xr-xr-x	1	root	bin	57348	May	11	1995	ftp
-rwxr-xr-x	1	root	bin	12288	May	11	1995	tty
-rwsr-xr-x	1	root	bin	8908	May	5	1995	umou
-rwxr-xr-x	1	root	root	14760	Feb	17	1995	ums

Das in der Spalte für die Zugriffsrechte angegebene erste Bit zeigt den Dateityp an. Es sind folgende Werte definiert:

d Directory

- normale Datei

l symbolischer Link

c zeichenorientierte Spezialdatei

b blockorientierte Spezialdatei

p Dateibesonderheit: fifo

Beispiel:

```
chmod 420 testdatei
```

Mit dieser Anweisung wird dem Benutzer das Leserecht und der Gruppe das Schreibrecht zugewiesen; der Benutzerklasse *others* wird überhaupt kein Zugriffsrecht zugeordnet.

```
r-- -w- ---
```


Eine alternative Methode für die Zugriffsrechteverteilung ist die sogenannte symbolische Zuordnung. Hier erhalten die Benutzerklassen ebenso wie die einzelnen Zugriffsrechte Buchstabenkürzel: u (*user*), g (*group*) und o (*others*). Sollten gemeinsam für alle Benutzergruppen Zugriffsrechte vergeben werden, so geschieht dies mit dem Buchstabenkürzel a (*all*). Diese Methode wird im Allgemeinen dann eingesetzt, wenn zu den bereits bestehenden Zugriffsrechten weitere ergänzt bzw. teilweise entzogen werden sollen. Hier gilt folgende Syntax:

```
chmod [ugoa] [+ -=] [rwx] [Objektname]
```

Beispiel:

```
chmod go+rw testdatei
```

Den bestehenden Zugriffsrechten werden für die Gruppe und für *others* die Lese- und Schreibrechte hinzugefügt.

Weitere UNIX-Anweisungen, die für eine Definition bzw. Manipulation von Zugriffsrechten verwendet werden, sind `unmask` oder `chown`.

HINWEIS

Bei der Vergabe von Zugriffsrechten muss darauf geachtet werden, dass die Schutzkonsistenz für Dateien und Verzeichnisse gewahrt bleibt, denn es ist durchaus möglich, Verzeichnissen andere Rechte zuzuordnen als den untergeordneten Dateiobjekten.

SVTX (Sticky Bit)

Normalerweise kann ein Benutzer in einem Verzeichnis, das ihm als »Eigentümer« zugewiesen ist, Dateien anlegen, lesen und auch wieder löschen. Haben diese von ihm generierten Dateien entsprechende Zugriffsrechte, so können diese Dateien auch von anderen Benutzern verändert oder gelöscht werden.

Ordnet man allerdings dem Verzeichnis, in dem sich die erstellte Datei befindet, durch die Anweisung `chmod +t [Verzeichnis]` ein sogenanntes *Sticky Bit* zu, so kann die Datei innerhalb des Verzeichnisses unabhängig von Zugriffsrechten lediglich vom Eigentümer (*owner*) gelöscht werden (der Oktalwert für das Sticky Bit ist 1000). Eine Datei mit den

Zugriffsrechten - *rwX rwX rwX* ist demnach nur vom Eigentümer zu entfernen, obwohl die Zugriffsrechte normalerweise einen anderen Benutzer ermächtigen, den Löschvorgang erfolgreich durchzuführen. Das übergeordnete Verzeichnis trägt dann folgende Zugriffsrechte: *drwx rwX rwt*, wobei das letzte Bit das *Sticky Bit* ist.

HINWEIS

Das allgemein zugängliche Verzeichnis */tmp* wird mit dem Sticky Bit versehen. Es kann durch die Zugriffsrechte *drwx rwX rwt* zwar von jedem Benutzer benutzt werden, allerdings kann ein Benutzer immer nur die eigenen Dateien wieder entfernen. Das versehentliche Löschen von Fremddateien wird somit verhindert.

SVTX (*Save Text Bit*) bewirkt – bei ausführbaren Dateien – eine bevorzugte Vorhaltung des jeweiligen Prozesses im Hauptspeicher (eine Art Cache). Bei einem erneuten Programmaufruf muss also nicht der gesamte Prozess wieder in den Speicher geladen werden, sondern es erfolgt ein direkter Zugriff auf den mit dem *Sticky Bit* versehenen Prozess. Durch ein zunehmend eingesetztes virtuelles Speichermanagement, das die Verweildauer von Prozessen im Hauptspeicher deutlich optimiert, hat diese Möglichkeit jedoch an Bedeutung verloren.

SUID (Set User Id) und SGID (Set Group Id)

Es gibt verschiedene Dateien (z.B. Systemdateien), die eines besonderen Zugriffsschutzes bedürfen. Sie werden daher automatisch dem Systemverwalter *root* als Eigentümer (*owner*) zugeordnet, mit entsprechenden Rechten versehen und sind somit für die anderen Benutzer nicht zu manipulieren. Gerade Systemdateien müssen jedoch regelmäßig auch von diesen Benutzern benutzt werden, um bestimmte Aktionen ausführen zu können, wie beispielsweise das Verändern des eigenen Kennworts durch das Kommando *passwd*. Da für die Manipulation von Kennwortdateien (*/etc/passwd*) Root-Berechtigung erforderlich ist, ein normaler Benutzer diese aber nicht besitzt, wird ihm durch das SUID-Bit lediglich für die Ausführungszeit des *passwd*-Kommandos die Root-Berechtigung erteilt. Allgemein formuliert bedeutet dies: Besitzt eine ausführbare Datei A das SUID-Bit, so erhält der jeweilige Benutzer für die Laufzeit des entsprechenden Prozesses die Berechtigung des Datei-Eigentümers. Dies führt dazu, dass auch Dateien manipuliert werden dürfen, die eigentlich einem anderen Benutzer (z.B. *root*) gehören – ein Umstand, der in einigen Situationen zwar erforderlich ist (Kennwortänderung), gleichzeitig aber auch ein nicht zu unterschätzendes Sicherheitsrisiko darstellt.

Während der Laufzeit des Prozesses besitzt der aufrufende Benutzer (bei Root-Berechtigung) volle Zugriffsrechte. Gelingt es ihm, ein Programm mit SUID-Bit, das eigentlich dem Systemverwalter (*root*) gehört, für seine Belange zu verändern, so kann er in einem UNIX-System durch den ungehinderten Zugriff auf gesicherte Root-Ressourcen erheblichen Schaden anrichten. Selbst die Tatsache, dass ein SUID auf Shell-Scripts nicht angewandt werden kann, ist durch die Integration in ein einfaches C-Programm zu umgehen.

Bei gesetztem SUID-Bit erhält der aufrufende Benutzer eine effektive *User-ID*, die lediglich für die Dauer des Programmlaufs gültig ist. Seine reale *User-ID* verändert sich in diesem Zeitraum nicht. So erhält die effektive *User-ID* während der Ausführung des *passwd*-Kommandos den Wert *euid* = 0 (0 ist der für den Benutzer *root* im System hinterlegte Wert), während die *ruid* = *nnn* unverändert bleibt und nicht überprüft wird. Allerdings gibt es Anweisungen, die eine Gegenüberstellung von *euid* und *ruid* vornehmen und nur bei völliger Identität entsprechende Manipulationen zulassen (beispielsweise das Kommando *mount*).

Mit der Eingabe *chmod u+s [ausführbare Datei]* erfolgt die Zuordnung des SUID-Bits. Entsprechend lautet die Eingabe

chmod g+s [ausführbare Datei] für das SGID-Bit, das analog zum SUID-Bit die Group-ID weitergeben kann.

ACL (Access Control List)

Ein weiteres Instrumentarium zur Sicherstellung restriktiver Zugriffsrechte bietet die Erweiterung von UNIX-Standardrechten durch *Access Control Lists* (ACLs). Durch die Funktionen *permit* (ausdrückliche Erlaubnis), *deny* (Verweigerung) und *specify* (Spezifikation) können innerhalb editierbarer ACLs erweiterte Zugriffsrechte definiert werden. Diese haben gegenüber den UNIX-Standardrechten absoluten Vorrang, wobei folgende Kommandos eingesetzt werden:

- `acledit`
Editieren der Zugriffsrechte und ihre syntaktische Überprüfung
- `aclput`
Übertragung von ACLs auf andere Dateiobjekte
- `aclget`
Lesen und Anzeige der ACLs für bestimmte Dateiobjekte

Sofern ACLs verwendet werden, erreichen sie normalerweise umfangreiche und damit äußerst unübersichtliche Ausmaße. Eine exakte Verwaltung ist dann meist nur noch sehr schwer

möglich. Darüber hinaus werden ACL-Informationen bei Datensicherungen, die mit den Kommandos *tar* und *cpio* durchgeführt werden, nicht übertragen. Lediglich *backup* und *restore* übernimmt diese sicherheitsrelevanten Informationen.

ACLs werden allerdings auch häufig im Zusammenhang mit Netzwerkkomponenten wie Routern oder Switchen eingesetzt. Entsprechende Konfigurationen ersetzen zwar keine komplexen Regelwerke, wie sie beispielsweise bei Firewalls zum Einsatz kommen, können jedoch in bestimmten Fällen eine Basissicherheit liefern. Manchmal reichen solche ACLs zur Erfüllung der Sicherheitsanforderungen bereits aus. Viel häufiger werden ACLs bei Switchen und Routern jedoch zur funktionalen Konfiguration eingesetzt (beispielsweise bei einer VPN-Konfiguration zwischen zwei Routern, die eine ausschließliche Kommunikation über das konfigurierte VPN zulässt und jede andere Kommunikation unterbindet).

Windows- und macOS-Zugriffsrechte

Trotz oft diskutierter Sicherheitslücken bei Windows-Betriebssystemen hat sich auch hier mittlerweile ein sehr umfangreiches Sicherheitssystem entwickelt, das zur Realisierung eines hohen Sicherheitsstandards verwendet werden kann. In vielen Fällen sind Vergleiche mit UNIX-

Systemen möglich. So können beispielsweise auch unter einem Windows-Serversystem einzelnen Dateiobjekten explizite Zugriffsberechtigungen zugeordnet werden. [Abbildung 7-2](#) zeigt beispielhaft unter Windows 11 die Berechtigungen, die einer Datei zugeordnet werden können.



Berechtigungen für "Gerhard Lienemann"	Zulassen	Verweigern
Vollzugriff	☒	☐
Ändern	☒	☐
Lesen, Ausführen	☒	☐
Lesen	☒	☐
Schreiben	☒	☐

Abb. 7-2 Sicherheitseinstellungen einer Datei unter Windows 11

Die Zugriffsrechteverwaltung auf macOS 12.4 (Monterey) sieht im Unterschied dazu so aus, wie [Abbildung 7-3](#) zeigt.

Du kannst lesen und schreiben	
Name	Zugriffsrechte
 gerdlienemann (I...	↕ Lesen & Schreiben
 staff	↕ Lesen & Schreiben
 everyone	↕ Lesen & Schreiben

Abb. 7–3 Sicherheitseinstellungen einer Datei unter macOS 12.4

Hier können die Zugriffsrechte für die einzelnen Benutzer durch die in der Spalte »Zugriffsrechte« der Drop-down-Liste individuell vergeben werden.

HINWEIS

Aufgrund der hohen Komplexität kann im Rahmen dieses Buches auf eine weitere Darstellung der Windows- und macOS-Zugriffsrechte nicht eingegangen werden. Hier gilt die Empfehlung weiterführender Literatur.

Benutzerauthentifizierung

Hinter dem Begriff *Benutzerauthentifizierung* verbirgt sich im Wesentlichen die Benutzeranmeldung über einen Login-Prompt. Das Login (Anmeldung) ist ein Anwendungsprogramm, das nach einer im System definierten

Authentifizierungsmethode vom Benutzer einen »einfachen Ausweis« verlangt. Dieser wird an Systemprozesse übergeben und mit gespeicherten Benutzerinformationen verglichen, bevor der Zugang zum System für den Benutzer geöffnet wird.

Normalerweise wird von einem Benutzer seine Kennung (*Username*) bzw. seine *User-ID* sowie ein Zugangskennwort verlangt. In einigen UNIX-Systemen kann eine Erweiterung dieser einfachen Authentifizierungsmethode vorgenommen werden, indem beispielsweise ein weiteres Kennwort angefordert wird. Leistungsfähige Systeme mit einigermaßen gut realisierten Sicherheitskonzepten lassen darüber hinaus auch besondere Kriterien für die Kennwortvergabe und deren Wartung zu (Kennwortmindestlänge, Kennwortwiederholungen, integrierbare Sonderzeichen, Kennwortablauf usw.).

Auf einem UNIX-System befinden sich in der Datei */etc/passwd* die relevanten Benutzerdaten, die zur Authentifizierung benötigt werden. Eine solche Datei könnte beispielhaft wie folgt strukturiert sein:

```
eva:FoMdWq20afN72:128:5:Benutzer Eva:/usr/eva/
```

```
gerd:Bpr2o2cy4HB4E:129:5:Benutzer Gerd:/usr/ge  
norbert:Hk.ABlAK4P53c:130:6:Benutzer Norbert:/  
willi:Anr994nHc6E30:131:6:Benutzer Willi:/usr/
```

Jedem Benutzer, der am System definiert ist, wird eine eigene Zeile der Kennwortdatei zugeordnet. Welche Felder dabei verwendet werden, ist in der folgenden Tabelle dargestellt:

Feldnummer	Bezeichnung	Beispielwert
1	Anmeldename (Kennung)	eva
2	Verschlüsseltes Kennwort	FoMdWq20afN72
3	User-ID (numerisch)	128
4	Gruppen-ID (numerisch)	5
5	Kommentar	Benutzerin Eva
6	Login-Verzeichnis	/usr/eva
7	Startprogramm	/bin/ksh

In den meisten UNIX-Implementierungen erfolgt eine Benutzerdefinition mit dem Kommando `/usr/sbin/adduser`. Im

Dialog werden die erforderlichen Benutzerinformationen abgefragt und in der Datei */etc/passwd* hinterlegt. Das Kennwort kann durch das Kommando */usr/bin/passwd* modifiziert werden. Die Datei */etc/group* definiert den Namen und die ID einer Gruppe und vermerkt außerdem alle Benutzer, die zu der entsprechenden Gruppe gehören.

Die R-Kommandos

Unter der Bezeichnung *R-Kommandos* oder *Berkley-Kommandos* leisten folgende drei Befehle einen wichtigen Beitrag zur Errichtung größtmöglicher Transparenz in einem UNIX-Netzwerk:

- *rloginremote* login
- *rshremote* shell
- *rcpremote* copy

Diese Anweisungen ermöglichen den Zugriff auf andere Systeme und damit auf deren Prozesse und Dateien, ohne dabei entscheidend restriktiven Schutzmechanismen zu begegnen. Die Erlaubnis, diese Kommandos zu benutzen, bedarf somit einer gründlichen Prüfung durch den System- bzw. Netzwerkverwalter.

Die Sicherheitsproblematik dieser Kommandos liegt darin begründet, dass für die Authentifizierung des aktiven Benutzers keine weiteren Kennwortprüfungen erforderlich sind. So werden lediglich Prüfungen auf Basis des eigenen Host-Namens bzw. der IP-Adresse sowie des Benutzernamens vorgenommen. Die entsprechenden sicherheitsrelevanten Dateien werden auf dem Rechner des Kommunikationspartners folgendermaßen definiert:

- Datei */etc/hosts.equiv*

Diese Datei wird auf dem Ziel-Host generiert und enthält eine Liste »vertrauenswürdiger« Hosts und Benutzer, die eine Ausführung der *Berkley-Kommandos* vornehmen dürfen. Es lassen sich allerdings auch »negative« Einträge vornehmen, d.h., der Zugriff kann bestimmten Rechnern bzw. Benutzern explizit untersagt werden.

- Datei *[\$HOME]/.rhosts*

Mit dieser Datei erhält jeder einzelne Benutzer auf dem Ziel-Host die Möglichkeit, Zugriffsrechte zu vergeben oder zu verweigern, da die *.rhosts*-Datei in seinem Login-Verzeichnis angelegt und dort ausgewertet wird. Der Zugriffsschutz über die *.rhosts*-Datei sollte alternativ zur Datei */etc/hosts.equiv* verwendet werden, da Zugriffe, die in der Datei erlaubt werden, in der *.rhosts*-Datei nicht wieder abgelehnt werden

können. Der Dateiaufbau entspricht dem der Datei */etc/hosts.equiv*.

rlogin

Das Kommando *rlogin* (*remote login*) startet eine Terminal-Sitzung auf einem entfernten Ziel-Host. Die Syntax dieses Kommandos lautet:

```
rlogin [options] [-l username] hostname
```

Eine sofortige Anmeldung ohne explizite Authentifizierung ist somit (unter Verwendung der Option *-l*) ohne Weiteres möglich, beispielsweise wie folgt: *rlogin -l willi kiel*. Eine in verschiedenen UNIX-Systemen mehr oder weniger umfangreiche Liste von Optionen (siehe nachfolgende Tabelle) kann dem Sicherheitsaspekt wieder (ein wenig) Relevanz verschaffen.

Option Beschreibung

- 8 Ermöglicht einen 8-Bit-Datenstrom.
- E Escape-Character werden ignoriert; zusammen mit der Option -8 wird eine absolut transparente Verbindung

realisiert.

- K KERBEROS-Authentifizierung wird abgeschaltet.
- d Schaltet das TCP-Socket-Debugging ein.
- e Spezifizierung des Escape-Characters (als Zeichen oder als Oktalwert `|nnn`)
- x Schaltet die DES-Verschlüsselung auf der *rlogin*-Session ein.

rsh

Die *Remote Shell* (*rsh*) führt auf einem entfernten Host ein Kommando aus. Dabei kopiert *rsh* seinen Standard-Input zum entfernten Kommando, das wiederum seinen Standard-Output an den Standard-Output des *rsh* weitergibt. Das *rsh*-Kommando wird normalerweise dann beendet, wenn auch der dadurch gestartete Prozess beendet ist. Die Syntax lautet:

```
rsh [options] [-l username] hostname [command]
```

Soll beispielsweise auf dem Ziel-Host *kiel* im Login-Verzeichnis des Benutzers *willi* eine Datei mit dem Namen *testdatei* angelegt werden, so muss folgende Eingabe erfolgen:

```
rsh -l willi kiel touch /home/willi/testdatei
```

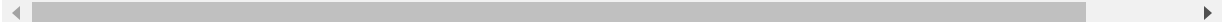
rcp

Die Anweisung *Remote Copy* (*rcp*) kopiert Dateien und Verzeichnisse zwischen zwei Systemen und verwendet dabei die Namenssyntax *filename@hostname.path*. Die vollständige Kommandosyntax lautet wie folgt:

```
rcp [options] source-file target-file
```

bzw. beim Kopieren von Verzeichnissen

```
rcp [options] [-r] source-file ... target-dire
```



Verwendbare Optionen sind in der nachfolgenden Tabelle dargestellt:

Option Beschreibung

- r Handelt es sich bei dem *source-file* um ein Verzeichnis, das komplett (samt interner Hierarchiestruktur) auf den Zielrechner kopiert werden soll, so ist diese Option anzugeben.
- p Ermöglicht die Beibehaltung von Time Stamp und Dateicharakteristika der Quelldatei während des Kopiervorganges. Die *umask*-Einstellung wird dabei ignoriert.
- x Schaltet die DES-Verschlüsselung auf der *rlogin*-Session ein.

Remote Execution (rexec)

Das Kommando *rexec* (*Remote Execution*) ist Bestandteil des TCP/IP und wird für die Jobausführung auf entfernten Rechnern verwendet. Es erfordert allerdings, anders als die Berkley-Kommandos, eine »normale« Benutzerauthentifizierung (wie bei *telnet* oder *ftp*). Die sicherheitskritische Komponente besteht hier in der Lesbarkeit des als Parameter übertragbaren Kennworts, sofern der sichere interaktive Dialog (Angabe von Benutzer und Kennwort) nicht verwendet werden soll. Für eine Jobautomation ist jedoch der mühsame Dialog oft ungeeignet, sodass die Sicherheitslücken

zugunsten komfortabler Funktionalität bewusst in Kauf genommen werden.

Ein einigermaßen sicherer Automatismus lässt sich allerdings über die Datei `[$HOME]/.netrc` realisieren. In dieser Datei werden auf der lokalen Maschine Ziel-Host, Benutzername und Kennwort hinterlegt, sodass der *rexec*-Client bei Aufruf diese Datei lesen kann und die entsprechend hinterlegten Werte bei Kontaktaufnahme dem *rexec*-Server übergibt. Die so vorgenommene Authentifizierung ergibt jedoch nur dann Sinn, wenn die lokale Datei `./netrc` vor unberechtigtem Zugriff geschützt werden kann; mit der Eingabe `chmod 600 $HOME/.netrc` lässt sich dies realisieren. Damit besitzt lediglich der Dateieigentümer Schreib- und Leserechte; alle anderen Benutzer haben an dieser Datei keinerlei Rechte. Die Datei `[$HOME]/.netrc` ist folgendermaßen aufgebaut:

```
machine [hostname] login [username] password [
```

Beispieleinträge:

```
machine berlin login eva password fahrrad
```

```
machine kiel login willi password kugel
```

```
machine muenster login gerd password gurti
```

Bei Eingabe des Befehls `rexec kiel ls -lat` wird lediglich überprüft, ob der angesprochene Ziel-Host mit dem autorisierten Benutzer (hier: *eva*) in der Datei `.netrc` übereinstimmt, bevor der Befehl `ls -lat` auf dem Ziel-Host ausgeführt wird. Die allgemeine Syntax des Kommandos `rexec` lautet wie folgt:

```
rexec hostname [-l username] [-p password]
```

```
[option] command
```

Externe Sicherheit

Die bislang erörterten Sicherheitsaspekte bezogen sich auf unternehmensspezifische interne Komponenten wie die

einzelnen Systeme oder den Rechnerverbund, also das lokale Netzwerk (LAN). Wenn auch einige der diskutierten Möglichkeiten (z.B. die *Berkley-Kommandos* oder die Standardrechte für UNIX- oder Windows-Systeme) in Bereichen netzwerkübergreifender Kommunikation eingesetzt werden können, so gehören sie doch zum klassischen Katalog interner Schutzmechanismen. Für die externe Sicherheit spielen andere Aspekte eine Rolle, die hauptsächlich den LAN-zu-LAN-Verbindungen über WANs (*Wide Area Networks*) Rechnung tragen. In der Unternehmenskommunikation erfolgt dabei die Auseinandersetzung mit einem besonderen Gefahrenmoment: dem Kontakt zu öffentlichen Netzwerken. Ob es lediglich um die Verbindung zweier verschiedener Standorte durch einfache Telefonleitungen geht, um den Aufbau eines vermaschten Router-Netzes oder die Anbindung an das Internet, es steht immer die Frage im Vordergrund: Wie können der sichere Datenfluss aus dem lokalen Standort hinaus und, vor allem, eine sichere dem Standort zufließende *Inbound*-Verbindung realisiert werden?

Öffnung isolierter Netzwerke

Über eines ist man sich bereits seit Längerem im Klaren: Die Zeiten der isolierten Kommunikation sind ein für alle Mal vorbei. Der Informationsbedarf eines Unternehmens mit

verschiedenen Standorten ist derart gewaltig, dass er nur noch dann befriedigt werden kann, wenn die benötigten Informationen in einer äußerst kurzen Zeit zur Verfügung gestellt werden können. Entscheidungen in nahezu allen Unternehmensbereichen erfordern den unmittelbaren Zugriff auf Informationen. Wartezeiten von mehreren Stunden oder gar Tagen können nicht in Kauf genommen werden, um die Flexibilität des Entscheidungspotenzials nicht zu gefährden.

Darüber hinaus spielt natürlich auch die Quantität von Informationen eine große Rolle. Die Beurteilung komplexer ökonomischer Sachverhalte und die Berücksichtigung relevanter Nebenbedingungen erfordern die Zusammenführung beträchtlicher Datenbestände aus unter Umständen verschiedenen Geschäftsfeldern und externen Datenquellen. Wenn hier eine zeitoptimierte Verfügbarkeit der Informationen gewährleistet werden muss, so ist dafür zu sorgen, dass die Netzwerksysteme im LAN, aber auch im WAN ausreichende Kapazitäten zur Verfügung stellen können.

Die Realisierung eines leistungsfähigen Kommunikationssystems für die Beschaffung und Weitergabe von Informationen darf gewiss nicht nur aus dem technischen Blickwinkel heraus betrachtet werden. Einen fast ebenso wichtigen Stellenwert nimmt hier die Konzeption einer

vernünftigen und praxisorientierten »Sicherheitspolitik« für die abgelegten Daten ein. Sie muss dafür sorgen, dass auf der einen Seite dem Sicherheitsaspekt ausreichend Rechnung getragen wird, andererseits aber die installierten Sicherheitsmechanismen den Datenfluss nicht unnötig behindern. Die Ausgewogenheit dieser im Grunde widersprüchlichen Ansätze gilt es zu finden. Die verschiedenen Bereiche externer Kommunikation lassen sich am einfachsten folgendermaßen klassifizieren:

- *LAN-WAN-LAN-Kommunikation*

Etablierung eines Unternehmensnetzwerks, das sich über mehrere Standorte oder Locations erstreckt. Die Integration externer unternehmensfremder Komponenten ist ausgeschlossen (sieht man einmal von der Benutzung gemieteter Leitungen entsprechender Provider ab).

- *Kommunikationskontakte zu Fremdfirmen*

Diese Anforderung ist immer damit verbunden, von extern auf die Datenbestände einer Firma zugreifen zu können. Diese Möglichkeit wird u.a. von Betreibern einschlägiger Wirtschafts- oder Wissenschaftsdatenbanken angeboten. Darüber hinaus soll die Öffnung der eigenen Systeme für Fremdfirmen ermöglicht werden, die auf bestimmte Bereiche der Unternehmensdaten zugreifen wollen.

- *Anbindung an öffentliche Netzwerke*

Diese Option stellt einem weltweiten Anwenderkreis (z.B. im Internet) Informationen und Daten zum wahlfreien Abruf zur Verfügung.

- *Cloud Computing*

Technisch nicht ganz neu, aber in zunehmendem Maße auch wirtschaftlich interessant ist das sogenannte »Cloud Computing«. Dabei werden verschiedene Dienstleistungen über öffentliche oder private Netzwerke angeboten (Datenspeicherung, Backup, Sharing-Dienste usw.), die über definierte Zugangspunkte das Unternehmensnetzwerk verlassen und u.U. weltweit an verschiedenen Orten gespeichert werden. Hier treten ganz besondere Sicherheitsfragen auf, wie beispielsweise rechtliche in Bezug auf den Ort der Datenspeicherung von Finanzdaten, aber auch eine ggf. fehlende Transparenz.

Das LAN/WAN-Sicherheitsrisiko

Die Sicherheitsvorsorge des unternehmenseigenen Netzwerks sollte spätestens zu dem Zeitpunkt abgeschlossen sein, wenn die ersten WAN-Verbindungen etabliert werden. Dadurch entstehen nämlich theoretische Möglichkeiten, in das Unternehmensnetz einzudringen und gegebenenfalls Schaden anzurichten. Bevor man jedoch in eine Sicherheitshysterie verfällt, sollten gründliche Überlegungen angestellt werden, wo

und vor allem wie die Datenbestände des Unternehmens gefährdet sein könnten.

Entscheidend für diese Frage ist: An welchen Punkten des internen Netzwerks existieren externe Ausgänge in andere Netzwerke? Um den Sicherheitsaufwand zu minimieren, ist es natürlich wünschenswert, die Anzahl der Ausgänge möglichst klein zu halten. Im Idealfall – im Hinblick auf die Sicherheit – wird lediglich ein einziger Rechner damit beauftragt, den externen Datenverkehr abzuwickeln, sodass lediglich an diesem »neuralgischen Punkt« (siehe [Abb. 7-4](#)) sicherheitsrelevante Maßnahmen getroffen werden müssen.

Diesem relativ statischen Gebilde steht oft die Notwendigkeit entgegen, für einen ausfallsicheren externen Kommunikationsbetrieb zu sorgen. Damit muss zumindest ein weiterer Rechner mit externen Zu- und Ausgängen zur Verfügung gestellt werden, der im Falle von Störungen auf dem Hauptsystem dynamisch dessen Aufgaben übernehmen kann (oder bei Bedarf manuell zugeschaltet wird).

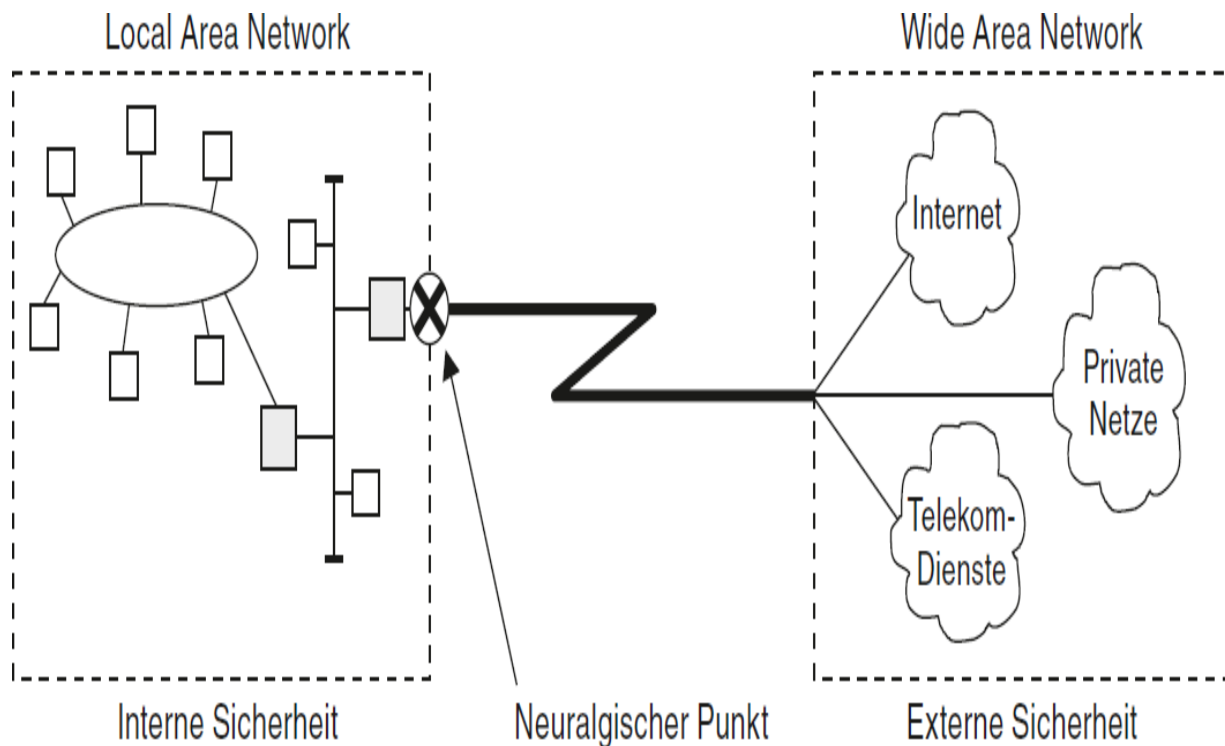


Abb. 7–4 Externe Netzwerke mit einem neuralgischen Punkt

Bei der Anbindung von Fremdfirmen ist das Sicherheitsrisiko nicht in dem Maße unkalkulierbar, wie es bei dem Zugang zu öffentlichen Netzwerken den Anschein hat. Einer Fremdfirma, die dem Unternehmen nicht bekannt oder in irgendeiner Weise verbunden ist, wird man den Zugang normalerweise nicht ermöglichen. Das Sicherheitsrisiko ist also überschaubar, sodass man hier mit Einzelmaßnahmen gezielt vorgehen kann. Eine sehr sichere Methode für eine solche Anbindung ist beispielsweise der beidseitige Einsatz von *Screening-Routern*, auf denen Paketfilter eingerichtet werden können. Dadurch kann gezielt die Verwendung bestimmter Anwendungen oder Protokolle untersagt werden. Heute werden in solchen Fällen

aber auch vermehrt kleine Firewall-Systeme eingesetzt, über die einfache VPN-Verbindungen etabliert werden und die somit volle Kommunikationskontrolle ermöglichen.

Organisatorische Sicherheit

Sieht man einmal von klassischen Bedrohungsszenarien ab, so gibt es eine Reihe weiterer Risiken, mit denen sich Unternehmen beschäftigen (bzw. beschäftigen müssten). Dabei handelt es sich um solche Risiken, die aus der eigenen Unternehmensstruktur und dem Benutzerverhalten entstehen. Beispiele hierzu sind:

- Data Leakage
- Nutzung potenziell gefährlicher Applikationen
- Prozessnetzwerke und ihr Schutz

Data Leakage

Ein Risiko, das Unternehmen direkt betrifft (einfach deshalb, weil es die eigenen Strukturen und Prozesse wiedergibt), liegt in der Art und Weise begründet, wie Unternehmensdaten gespeichert und kommuniziert werden. Auch wenn es oft Richtlinien gibt, die das Speichern und Archivieren von Unternehmensdaten konkret regeln (beispielsweise das in den Richtlinien vorgeschriebene Speichern auf Servern), wird in

der Praxis davon oft abgewichen. Reisende Mitarbeiter nutzen für ihren mobilen Arbeitsplatz USB-Sticks, portable Festplatten oder speichern ihre Arbeitsdokumente einfach auf der eigenen lokalen Festplatte ab.

Aber auch dann, wenn Benutzer ihren Arbeitsplatz nicht verlassen, wird häufig vom lokalen Speichermedium Gebrauch gemacht, oft auch einfach aus Bequemlichkeit (wer würde nicht den schnellen Zugriff auf lokal verfügbare Dokumente gegenüber den Dokumenten auf Netzlaufwerken vorziehen – der Ladevorgang ist oft um ein Vielfaches langsamer). Eine »Unternehmenspolizei«, die die Einhaltung von Richtlinien überprüft, gibt es in der Regel nicht – und in den Aufgaben der Audit- oder Security-Teams sind solche Prozeduren normalerweise nicht enthalten.

Es sind aber gerade die Server, auf denen – entsprechende Regeln und Prozesse vorausgesetzt – eine konsequente Umsetzung von »Data Leakage«-Sicherheitsrichtlinien möglich ist. Der »Abfluss unternehmenskritischer Daten« (*Data Leakage* – Datenlecks) stellt ein nicht zu unterschätzendes Risiko dar, dessen sich die meisten Unternehmen nicht bewusst sind oder wenn sie nach dem Motto verfahren: »Warum sollte so etwas gerade bei uns passieren?« Es verfestigt sich dann eine Haltung

wie: »Die Wahrscheinlichkeit eines Datenklau ist bei uns äußerst gering.«

Dabei können geeignete Vorgaben bei der Speicherung und Archivierung von Daten sehr effektiv gegen diesen illegalen Datenabfluss eingesetzt werden. Lässt man beispielsweise die Speicherung sensibler Entwicklungsdaten nur auf bestimmten Servern zu und sorgt dort ganz gezielt für einen ausgesprochen hohen Sicherheitsstandard (starke Authentifizierung, Zertifikate, Vier-Augen-Prinzip, Verschlüsselung usw.), so bleiben auch die entstehenden Kosten in einem vertretbaren Rahmen.

Nutzung potenziell gefährlicher Applikationen

Wer hat nicht schon einmal versucht, die im Hausgebrauch gern genutzte Kommunikationssoftware *Skype* auf dem eigenen Firmenrechner zu installieren und somit auch die bequeme Möglichkeit einer kostenlosen Kommunikation mit Daheimgebliebenen zu nutzen? Klar funktioniert das prima, und sicher ist dies auch ganz im Sinne einer kostengünstigen Kommunikation (schließlich muss man sich dann nicht immer wieder für die mehr oder weniger umfangreichen Telefonate mit dem Firmenhandy rechtfertigen). Aber ist das auch sicher?

Den meisten Benutzern ist es nicht bewusst (das gilt freilich auch für den Betrieb der privaten Rechner zu Hause), welchen Sicherheitsrisiken ihr Rechner ausgesetzt ist, wenn eine solche Software wie Skype genutzt wird. Mittlerweile gibt es allerdings eine Vielzahl ähnlicher Software, die ein vergleichbares Risiko beinhaltet. Da Skype im Hause Microsoft zunehmend durch die wesentlich sicherere und komplexere Software Teams ersetzt wird (sowohl für geschäftliche Zwecke als auch im privaten Bereich), wurde einem Teil der Sicherheitsbedenken entgegengewirkt.

Im vorherigen Abschnitt wurde auf das Risiko »Data Leakage« hingewiesen. Gerade »Skype« oder »Teams« könnte dieses Risiko deutlich verstärken – und das paradoxerweise durch die Nutzung von Sicherheitstechniken wie »Verschlüsselung«. Denn wer weiß schon, was ein Benutzer im Unternehmensnetzwerk via verschlüsseltem Dateitransfer an Daten wohin schickt? Kein noch so intelligentes Content-Scanning-System vermag in diesen Datenfluss hineinzuschauen.

Wie so oft ist »geeignete Kontrolle« das Zauberwort. Ist man in der Lage, die Kommunikation solcher Anwendungen zu kontrollieren, führt das zu einer weitgehenden Entschärfung des Risikos. Im [Abschnitt 7.7.7](#) wird auf einen alternativen

Sicherheitsansatz hingewiesen, der dem illegalen Abfluss kritischer Daten entgegenwirken kann.

Prozessnetzwerke und ihr Schutz

Eine ganz andere Kategorie von Sicherheitsrisiken stellen besondere Netzwerke im prozesstechnischen Umfeld dar. Hier sind in der Regel produzierende Betriebe betroffen, die einen ganzen Park spezieller Systeme bei der Herstellung aufwendiger Produkte einsetzen müssen. Diese Systeme sind oft so speziell, dass sie den Sicherheitsstandards eines Unternehmens hinsichtlich Betriebssysteme und Anwendungssoftware (Sicherheitspatches) nicht entsprechen und auch nicht entsprechen dürfen, um ihre Funktionalität und Stabilität nicht zu gefährden.

Man stelle sich vor, die Software zum Betrieb der Robotersysteme einer Automobilproduktion würde mit einem Sicherheits-Patch versehen, der unter dem aktuellen Betriebssystem nicht funktioniert und zu ihrem Ausfall führt. Der Schaden (Produktionsausfall) wäre sicherlich enorm.

Dabei wäre dieses Problem nicht aufgetreten, hätte man für den Betrieb der Produktionsroboter eine Standardsoftware und ein Standardbetriebssystem eingesetzt, das man vorher ausreichend hätte testen können. Aber leider haben sich in den

Jahren die Produktionsnetze von Unternehmen oft unabhängig voneinander entwickelt und ihren Fokus auf die Anwendung selbst gerichtet, nicht aber auf den Prozess der Integration in die übrige IT-Infrastruktur. Ein Risiko, das nur langfristig beseitigt werden kann.

Angriffe aus dem Internet

Mit der Öffnung von Unternehmensnetzwerken zum Internet beginnt in der Regel die Diskussion um das Maß an erforderlicher Sicherheit, das zur Absicherung der eigenen Unternehmensressourcen benötigt wird. Insbesondere stellen sich in diesem Zusammenhang folgende Fragen bzw. sind folgende Aussagen zu finden:

- Wie viel Sicherheit ist nötig?
- Können wir uns die benötigte Sicherheit leisten?
- Wird die Gefahr von Angriffen nicht drastisch überschätzt?
- Da wir für Hacker/Cracker nicht interessant sind, ist es eher unwahrscheinlich, dass wir »angegriffen« werden ...
- Wir lassen es einfach darauf ankommen – dann können wir ja immer noch reagieren!

Solche Fragen bzw. Aussagen werden stets in die Diskussion geworfen, wenn die Angriffsgefahr auf

Unternehmensressourcen aus dem Internet thematisiert wird. Dabei ist es relativ einfach, eine vernünftige Haltung zu sinnvollen Sicherheitsmaßnahmen zu entwickeln. Niemand wird bestreiten, dass es keinen absoluten Risikoausschluss bei der Netzwerkabsicherung gibt. Selbst bei progressivem Investitionsverlauf wird es eine hundertprozentige Sicherheit nicht geben. Es gilt daher, ein ausgewogenes Verhältnis zwischen Kosten und Sicherheit herzustellen, um somit den »normalen« Sicherheitsanforderungen zu genügen. Dabei wird die Definition des »ausgewogenen Verhältnisses« bei einer Bank oder Versicherung sicher anders ausfallen als bei einem kleinen Unternehmen des Einzelhandels.

Das Thema Sicherheit im Netzwerk muss ernst genommen werden. Allerdings ist dazu ein vernünftiges Augenmaß für eine sinnvolle Finanzierung erforderlich, denn Sicherheit ist niemals Selbstzweck, sondern muss im Kontext der Anforderungen gesehen werden.

»Hacker« und »Cracker«

In den Medien wird der Begriff »Hacker« meist als Sammelbegriff für jeden Computerfanatiker verwendet, der in fremde Netzwerke und Systeme eindringt und dort Schaden anrichtet. Somit hat dieser Begriff ein eindeutig negatives

Image erhalten, was ihm jedoch nicht gerecht wird. Vielmehr steht hinter einem »Hacker« zumeist ein System- und Kommunikationsexperte, der in zum Teil mühevoller Detailarbeit im positiven Sinne Schwachstellen aufdeckt und somit zur Schließung von Sicherheitslücken beiträgt.

Unternehmen gehen heute zunehmend dazu über, ihr Sicherheitsteam mit solchen »Hackern« zu ergänzen, damit sie bereits proaktiv agieren und unter Verwendung des Wissens und der Fähigkeiten dieser Experten das Sicherheitsrisiko ihrer Ressourcen minimieren können.

Für den »Experten« mit krimineller Energie hat sich im Laufe der Zeit der Begriff »Cracker« (abgeleitet von »criminal hacker«) etabliert. In den Medien wird er leider nur sehr selten verwendet, sodass es hier schnell zu Verunglimpfungen der eigentlichen *Security Task Force*, den Hackern, kommt. Jenen eindeutig auf Zerstörung oder persönlicher Bereicherung ausgerichteten »Crackern« wird oft ungerechtfertigterweise das glorifizierende Bild eines technisch hochbegabten Computerexperten zugewiesen. Letztendlich handelt es sich aber lediglich um einen Kriminellen, der auf einem speziellen Gebiet seine Straftaten begeht.

Sobald das Eindringen in ein fremdes System ohne die Erlaubnis des verantwortlichen Sicherheitsexperten geschieht, handelt es sich um einen illegalen Vorgang. Wird hierzu jedoch eine ausdrückliche Genehmigung erteilt, so handelt es sich in der Regel um einen expliziten Sicherheits-Check. Einige Unternehmen haben sich mittlerweile darauf spezialisiert, sogenannte »Penetration-Tests« durchzuführen. Dabei werden speziell entwickelte und mit dem Auftraggeber abgestimmte Angriffsszenarien durchgespielt und gegen die externen Zugangspunkte eines Unternehmensnetzwerks gerichtet (meist Firewalls oder Access-Router).

Scanning-Methoden

Entgegen der verbreiteten Meinung, »Cracker« seien Einzeltäter, die ihre Aktionen spontan durchführen, sieht die Realität oft anders aus. Bevor gezielte Angriffe auf ein bestimmtes Unternehmensnetzwerk durchgeführt werden, erfolgt eine umfangreiche Analyse des »Objekts der Begierde«. Somit kann sichergestellt werden, dass keinerlei relevante Informationen übersehen werden und das Ausspionieren oder die Sabotage bestimmter Unternehmensressourcen erfolgreich ist. Ein beliebtes Werkzeug zur Ausspionierung von Rechnernamen eines fremden Netzwerks ist beispielsweise NSLOOKUP. Es liefert umfangreiche Daten über IP-Adressen

und Rechnernamen einer DNS-Zone. Oft werden auch sogenannte »sprechende« Namen für die Bezeichnung von Rechnern verwendet (z.B. *fw.test.de*; hier könnte es sich bei *fw* um eine Firewall handeln), was natürlich die Interpretationsarbeit des Crackers deutlich vereinfacht. Hat man erst einmal die interessanten Rechner ausfindig gemacht, kann man damit beginnen, diese systematisch auszukundschaften.

Methode 1: ICMP-Requests

Eine interessante Zusatzinformation liefert das unter UNIX verfügbare Hilfsprogramm *icmpquery*. Es ist auf zahlreichen Servern im Internet als Source-Datei (*icmpquery.c*) verfügbar. Neben der Abfrage von Uhrzeit (hier wird der *ICMP-Typ 13/timestamp* verwendet) liefert es beispielsweise die folgende Anweisung

```
icmpquery -m 195.233.122.1
```

```
195.233.122.1: 0xFFFFFFFF0
```

die Subnetzmaske (hier in dezimaler Schreibweise: 255.255.255.240), in der die IP-Adresse 195.233.122.1 »beheimatet« ist, und gibt somit einen Teil der Struktur des adressierten IP-Netzwerks preis.

Methode 2: Port-Scanning

Eines der verbreitetsten Verfahren zur Ausspionierung von Rechnern (insbesondere Servern) ist das sogenannte *Port-Scanning*. Hierbei handelt es sich um die Analyse der auf dem Zielsystem aktiven TCP- und UDP-Server-Prozesse, um damit Rückschlüsse auf die Funktion des Servers zu ziehen.

Ferner ist es für einen Angreifer äußerst interessant, welche aktiven Prozesse den Zustand *listening* besitzen, um diese gegebenenfalls für eigene Aktivitäten zu missbrauchen (der Angreifer »hängt« sich quasi in die halb offene Verbindung ein und kann somit aktiv eine Kommunikation auf dem Zielsystem initiieren). Eine solche Information erhält man beispielsweise, wenn ein *TCP-SYN-Scan* durchgeführt wird. Dabei sendet der Angreifer ein TCP-Paket mit aktivem SYN-Flag. Erhält er daraufhin ein SYN/ACK-Paket, so befindet sich der Ziel-Port im Status *listening*; besteht die Antwort jedoch aus einem RST/ACK-Flag, so ist der Ziel-Port voraussichtlich nicht aktiv.

Die wirksamste Maßnahme, *Port-Scans* und Ähnliches zu unterbinden bzw. einzuschränken, ist die Deaktivierung aller Ports, die für den Betrieb des Systems nicht unbedingt benötigt werden. Eine direkte Verhinderung des *Port-Scanning* selbst ist nicht möglich, wohl aber die Information, dass ein solcher Vorgang durchgeführt wird bzw. wurde. Es gibt eine Reihe von Hilfsprogrammen, die diese Informationen liefern können.

Eine weitere wirkungsvolle Maßnahme ist der Einsatz von IDS-Systemen (IDS = *Intrusion Detection System*), die solche und andere Angriffe erkennen können. Sogenannte IRS-Systeme (IRS = *Intrusion Response System*) bieten darüber hinaus die Möglichkeit, bei Erkennung von Angriffen geeignete Gegenmaßnahmen einzuleiten. Heutige Firewall-Systeme bieten meist ähnliche Funktionen zur »Intrusion Prevention« und machen bei geringeren Anforderungen den Einsatz umfangreicher IDS- bzw. IRS-Systeme oft überflüssig.

Methode 3: Betriebssystemanalyse

Um die Entscheidung zu treffen, welches System ausspioniert werden soll, ist die Kenntnis um das Betriebssystem des Zielrechners natürlich von entscheidender Bedeutung. Jedes Betriebssystem hat seine charakteristischen Eigenschaften und

vor allem Schwachstellen, auf die sich ein potenzieller Angreifer einrichten muss.

Das Hilfsprogramm, das wohl die genauesten Ergebnisse liefert, trägt den Namen *nmap*. Es ist auf verschiedenen Servern im Internet verfügbar und lässt sich beispielsweise in der Linux-Version über das Paket-Installations-Tool installieren.

Denial of Service Attack

Eine der größten Gefahren für Server im Internet ist eine sogenannte *Denial of Service Attack*, kurz DoSA genannt. Bei diesem Angriffsmuster werden unter Verwendung verschiedener Methoden die Zielsysteme mit Serviceanfragen »bombardiert« und so lange unter Beschuss gesetzt, bis sie schließlich aus Ressourcenmangel nicht mehr funktionsfähig sind und neu gestartet werden müssen.

Eine Variante dieser klassischen DoSA stellt die bewusste »Überflutung« einer Verbindung mit geringer Bandbreite mit Daten aus einer Verbindung mit sehr hoher Bandbreite dar. Soll beispielsweise die permanente Internetverbindung eines Unternehmens mit einer Bandbreite von 128 kbit/s zur völligen Auslastung gebracht werden, so ist das aus einer Verbindung mit wesentlich größerer Kapazität, z.B. 2 Mbit/s, leicht möglich.

Abbildung 7-5 zeigt diesen Effekt in einer veranschaulichenden Grafik.

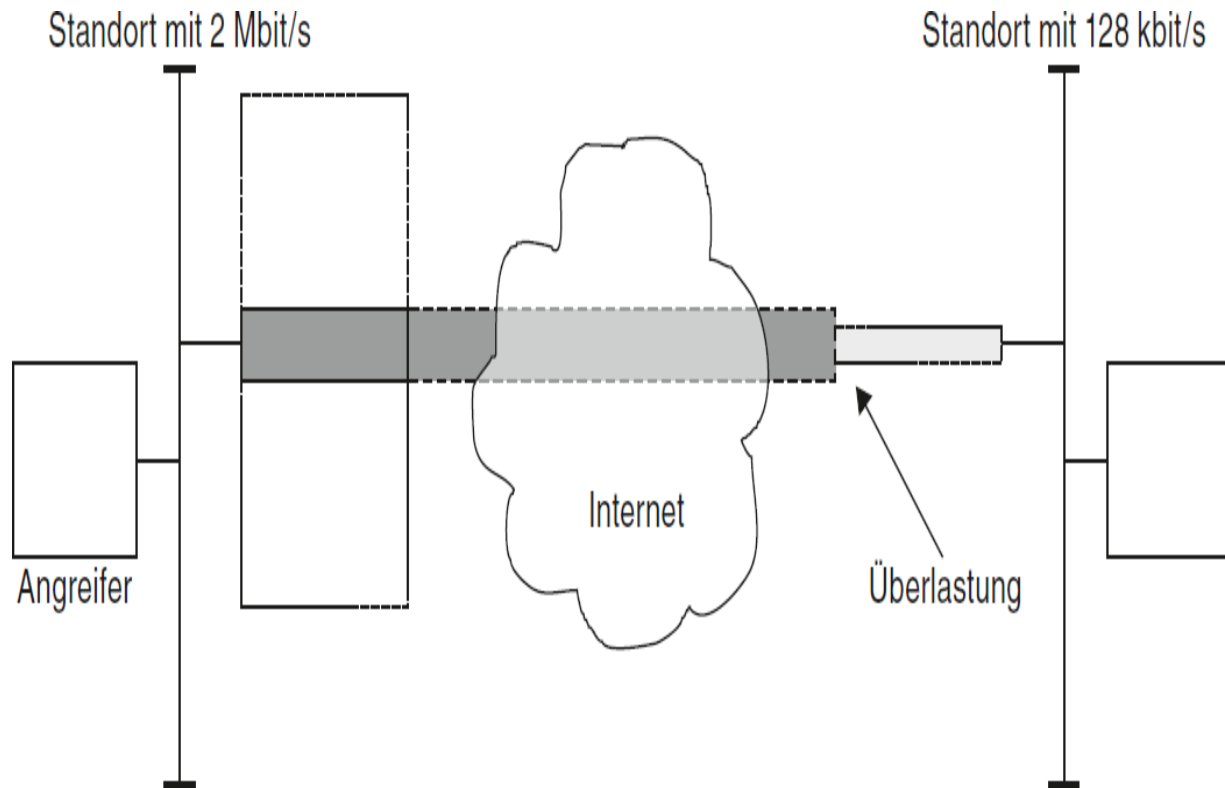


Abb. 7-5 *Beispiel für manipulierten Bandbreitenverbrauch*

In dieser Abbildung wird ein Datenstrom erzeugt, der gerade einmal 5% der 2-Mbit/s-Leitung (links) belegt. Kommt dieser *Traffic* jedoch bei dem Zielsystem an, das lediglich über eine 128-kbit/s-Verbindung verfügt, so ist diese zu 75% ausgelastet. Erhöht man diesen Datenstrom nur geringfügig auf 8%, so liegt bereits eine Überlastung der Verbindung des Zielsystems vor, was dazu führt, dass es völlig blockiert.

Verteilte DoSA

Eine besonders wirkungsvolle DoSA hat in der Vergangenheit die Webserver einiger renommierter Internetfirmen für mehrere Stunden lahmgelegt. Es handelte sich dabei um eine DDosA, eine *Distributed Denial of Service Attack*, bei der die Angriffe von zahlreichen zuvor manipulierten Angriffssystemen geführt wurden.

Die beiden grundsätzlichen Schwachstellen bei DDoSAs liegen einerseits in der Fälschung von gesendeten Datenpaketen (*IP-Spoofing* – wie auch bei einfachen DoSAs) und andererseits darin, dass auf den angegriffenen Zielsystemen für diesen Zweck hergestellte Programme gezielt platziert worden sind. In den meisten Fällen werden diese »implantierten« Programme überhaupt nicht bemerkt. Beginnt dann der Angriff, so wird er nicht nur von einem einzigen, sondern von zahlreichen Angreifern geführt, was die Effektivität natürlich deutlich erhöht.

Die Vorbereitung dieser Angriffe wird sorgfältig geplant. Es werden genau diejenigen Systeme lokalisiert, die nicht einmal mit Grundschutzmaßnahmen (z.B. Schutz durch eine Firewall) ausgestattet sind, sodass ein Eindringen in diese Systeme keinen größeren Aufwand erfordert. In einem nächsten Schritt

werden dann über sogenannte »Trojanische Pferde« die erforderlichen Angriffsprogramme installiert. Der eigentliche Angriff wird durch den Hauptangreifer und ein zentral verteiltes Signal ausgelöst (siehe [Abb. 7-6](#)).

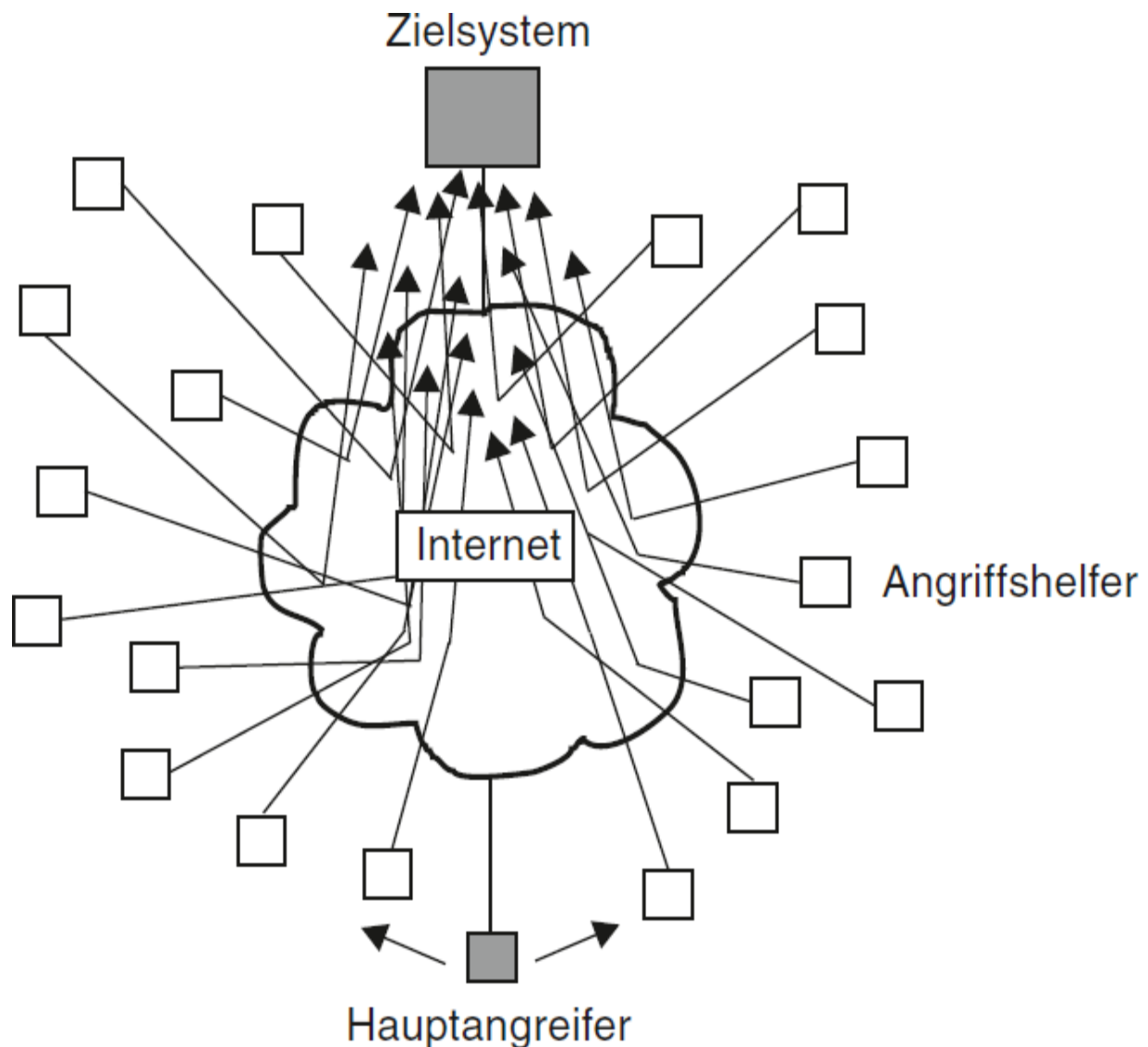


Abb. 7-6 *Verteilte Denial of Service Attack (DoSA)*

HINWEIS

»Trojanische Pferde« oder auch kurz »Trojaner« sind Programme, die über Downloads oder als Mail-Attachment angeboten eine völlig andere Funktion vorgeben (z.B. Spiele), als es ihr eigentlicher Zweck ist.

Die Komplexität des dargestellten Angriffsszenarios macht deutlich, dass hier nur durch eine gemeinsame Vorgehensweise der verschiedenen Zielgruppen wirkungsvolle Gegenmaßnahmen durchgeführt werden können. Hier einige Beispiele:

- Betreiber von Servern:
 - Einsatz von Paketfiltern
 - Automatische Angriffserkennung durch Einsatz sogenannter *Intrusion Detection Systems*
 - Aufstellung von Notfallplänen (*Contingency Plan*)
 - Serverabsicherung
 - Restriktive Vergabe von Rechten
 - Protokollierung
- Betreiber von Netzwerken:
 - Implementierung von Anti-Spoofing-Mechanismen
 - Automatische Angriffserkennung durch Einsatz sogenannter *Intrusion Detection Systems*
- Endanwender:

- Automatische Angriffserkennung durch den Einsatz lokaler *Intrusion Detection Systems* bzw. von lokalen »Firewalls«
- Rekonfiguration der Internetbrowser hinsichtlich restriktiver Sicherheitseinstellungen (Deaktivierung aktiver Inhalte wie ActiveX; es ist allerdings anzunehmen, dass Microsoft das Ende von ActiveX einläutet, da der neuere Microsoft-Internetbrowser »Edge« keine ActiveX-Implementierung mehr enthält)

SYN-Flooding

Ein sehr wirkungsvoller *Denial-of-Service-Angriff* stellt das sogenannte *SYN-Flooding* dar. Zur Darstellung seiner Charakteristik muss die TCP-Verbindung etwas genauer untersucht werden.

Eine TCP-Session wird ja bekanntermaßen im sogenannten *Three Way Handshake*-Verfahren aufgebaut:

- *Phase 1:*

Der Session-Initiator signalisiert seinem Kommunikationspartner durch ein gesetztes SYN-Flag (eines der fünf möglichen Flags im TCP-Frame-Header), dass er eine TCP-Session aufbauen möchte.

- *Phase 2:*

Der Kommunikationspartner empfängt den Frame mit dem gesetzten SYN-Flag und antwortet mit einem zusätzlich gesetzten ACK-Frame. Dies bedeutet, dass er dem Verbindungswunsch entsprechen will.

- *Phase 3:*

Der Verbindungs-Initiator empfängt die SYN/ACK-Flag-Kombination und bestätigt den Verbindungsaufbau schließlich mit einem TCP-Frame mit gesetztem ACK-Flag. Die TCP-Session ist nunmehr aktiv.

Dieses Drei-Phasen-Konzept macht sich der Angreifer beim *SYN-Flooding* zunutze und startet seinen Angriff zunächst mit einer falschen IP-Quelladresse (*Spoofing*). Das Zielsystem akzeptiert – wie im beschriebenen Normalfall – die Anfrage (*Request*) zum Verbindungsaufbau und sendet dem nichtexistierenden Verbindungs-Initiator (dem Angreifer) den TCP-Frame mit gesetztem SYN/ACK-Flag. Erhielte diesen Frame ein tatsächlich existierendes System, so würde es mit einem RST-Flag antworten und damit den Prozess des Verbindungsaufbaus abbrechen, da ja ein entsprechender Request nicht initiiert wurde. Da es aber keinen tatsächlichen Empfänger des SYN/ACK-Flags gibt, bleibt dieser TCP-Frame unbeantwortet.

Bis zur Erreichung einer Timeout-Schwelle (diese kann wenige Sekunden, aber auch viele Minuten betragen) bleibt diese TCP-Verbindung im SYN-RECV-Status geöffnet und belegt damit den zugeteilten Speicher des Zielsystems. Der weitere Verlauf des *SYN-Flooding* ist jetzt absehbar: Der Angreifer wiederholt diesen Vorgang mehrere Male und blockiert somit allmählich die verfügbaren Systemressourcen bis zur völligen Erschöpfung. Das Zielsystem ist nun nicht mehr einsetzbar und kann erst nach einem Neustart wieder verwendet werden. Geschieht dies mit wichtigen Servern, die geschäftskritische Anwendungen zur Verfügung stellen, so kann ein solcher Ausfall einen erheblichen Schaden zur Folge haben.

Ping of Death

Das Problem des *Ping of Death* (PoD) stellt sich folgendermaßen dar: Die Paketlänge eines IP-Frames darf die Gesamtgröße von insgesamt 65.535 Bytes nicht überschreiten. Dabei ist die Header-Länge von 20 Bytes bereits enthalten. Bei einem Ping-Kommando kommen zusätzlich 8 Bytes hinzu, die für den ICMP-Header des *ICMP Echo Request* benötigt werden. Netto bleiben somit $65.535 - 20 - 8 = 65.507$ Bytes übrig. Gelingt es einem Angreifer, ein Datenpaket zu senden, das trotz Fragmentierung größer ist als jener Schwellenwert (verstärkt wird dies durch zusätzliche Einträge im IP-Optionenfeld), so

kann es zu einem Pufferüberlauf kommen, da die fragmentierten Pakete ja erst dann zusammengesetzt werden, wenn sie alle am Zielort angekommen sind. Dieser Effekt geht dann oft mit einem Systemabsturz einher, sodass ein Neustart erforderlich wird.

In jüngster Vergangenheit ist dieses Risiko allerdings durch intensives Aktualisieren der Schwachstellen in Betriebssystemen (*Patching*) reduziert worden.

DNS-Sicherheitsprobleme

Die Konfiguration eines DNS-Servers ist in der Regel nicht besonders kompliziert, sie birgt aber einige sicherheitsrelevante Tücken, die es zu beachten gilt. Nachfolgend sollen einige wesentliche, offensichtliche Sicherheitslücken erläutert werden, die jedoch in der Regel recht problemlos geschlossen werden können.

Eingeschränkte Zonentransfers

Eine »offene« Konfiguration von Nameservern, die es jedem beliebigen DNS-Server erlaubt, via Zonentransfer die eigene DNS-Datenbasis zu übertragen, ist sicherheitskritisch. Die Daten innerhalb eines Zonentransfers sind für externe Angreifer mitunter sehr interessant, da sie Aufschluss über die Funktion

wichtiger Server des Netzwerks liefern. Hier macht sich das Verwenden sogenannter »sprechender Namen« negativ bemerkbar, da nun auch unternehmensfremde Personen Einblick in die eigenen Rechner nehmen können. Daraus entstehen dann Überlegungen, ob es sich lohnt, einen solchen Rechner zu attackieren oder nicht. Wird beispielsweise eine Firewall auch mit dem Namen »firewall« oder ein SAP-Rechner mit »sapserv1« bezeichnet, so sind gleich die lohnenden Angriffsobjekte identifiziert, und der »Angriff« kann beginnen.

Um die »Geschwätzigkeit« des eigenen DNS-Servers zu vermeiden, sollte man die notwendigen Zonentransfers und Weiterleitungen nur zu vertrauenswürdigen Servern zulassen.

HINWEIS

Entsprechende Einschränkungen lassen sich in der Regel für die DNS-Implementierungen beliebiger Betriebssysteme vornehmen. So lautet beispielsweise unter UNIX/Linux innerhalb der Konfigurationsdatei *named.conf* die entsprechende Option *allow-transfer*, in der diejenigen Rechner angegeben werden, die für einen Zonentransfer zugelassen sind.

DNS-Spoofing

Beim *DNS-Spoofing* wird versucht, DNS-Servern falsche IP-Adressen zu existierenden Domainnamen unterzuschieben. Dadurch erfolgt bei der Verwendung bestimmter DNS-Host-Namen die Umsetzung auf diese falschen IP-Adressen und führt zu IP-Verbindungen zu Fremdsystemen des Angreifers. Das *DNS-Spoofing* läuft folgendermaßen ab:

- *Phase 1:*

Der Angreifer www.attack.de sendet eine DNS-Query an das Zielsystem dns.ziel.de, in der er die IP-Adresse eines Namens aus seiner eigenen Domäne anfragt.

- *Phase 2:*

Das Ziel-DNS-System kann diese Anfrage natürlich nicht beantworten und richtet diese an den DNS-Server dns.attack.de des Angreifers (er ist ja für die angefragte Domain zuständig).

- *Phase 3:*

Der Angreifer übernimmt aus der empfangenen DNS-Query die eindeutige *qid* (*Query Identification*), aus der die *qid* der nächsten Anfrage des angegriffenen Systems ermittelt werden kann. Diese wird später für eine Manipulation benötigt (Phase 6).

- *Phase 4:*

Nun wird eine IP-Adresse (z.B. www.opfer.de) aus der Domain des angegriffenen Systems opfer.de bei dem DNS-Zielsystem dns.ziel.de angefragt.

- *Phase 5:*

Das DNS-Zielsystem kann natürlich diese angefragte IP-Adresse auch nicht unmittelbar liefern, sondern muss dazu den DNS-Server des angegriffenen Systems dns.opfer.de befragen (das System benutzt ja dazu die in Phase 3 ermittelte *qid*).

- *Phase 6:*

Bevor dann der kontaktierte DNS-Server dns.opfer.de antworten kann, wird dem DNS-Zielsystem (dns.ziel.de) vom Angreifer die manipulierte DNS-Antwort (*Response*) geschickt, die die IP-Adresse des angreifenden Systems www.attack.de enthält. Später eintreffende Antworten (der DNS-Server dns.opfer.de antwortet ja ebenfalls, nur etwas später) werden verworfen.

Kommuniziert anschließend jemand mit dem Rechner des angegriffenen Systems (www.opfer.de), so nimmt er aufgrund der »untergeschobenen« IP-Adresse des Angreifers mit dem Rechner www.attack.de Kontakt auf, und dieser kann sämtliche ursprünglich für den Rechner www.opfer.de bestimmten Daten abfangen.

HINWEIS

Die Unzulänglichkeit bei der *qid*-Administration wird von neueren DNS- bzw. BIND-Versionen durch wiederholte DNS-Abfragen vermieden. Erst bei Übereinstimmung der zweiten mit der ersten DNS-Response wird die so ermittelte IP-Adresse in die DNS-Datenbasis aufgenommen und gespeichert. Es ist daher besonders wichtig, ältere DNS-Server mit den aktuellen Software-Upgrades zu versehen, damit diese Sicherheitslücke geschlossen wird.

Zum Thema DNS sind in den letzten Jahren allerdings einige Veröffentlichungen zur Verbesserung der Sicherheit erfolgt, die in Betriebssystemen bereits implementiert wurden. Es wurden auch einige RFCs zu dem Thema erstellt (beispielsweise der RFC 9103 vom August 2021 »DNS Zone Transfer over TLS«, in dem die verschlüsselte Übertragung von Zonendaten beschrieben wird anstelle der sonst üblichen unsicheren Übertragung in Klartext).

Schwachstellen des Betriebssystems

Auch wenn jedes Betriebssystem, sei es im administrativen Umfeld oder in rein technischen Umgebungen, unterschiedliche Charakteristika aufweist, so sollte der wichtigste Grundsatz

lauten: Unmittelbar nach Veröffentlichung sofortiges »Patching« der Systeme!

Damit wird verhindert, dass die nach Bekanntwerden von Schwachstellen betroffenen Betriebssysteme nicht den potenziellen Angriffen von Schad-Code ausgesetzt wird und somit Unternehmensdaten (natürlich auch die von Privatpersonen) manipuliert, gelöscht oder gestohlen werden.

Auf konkrete Maßnahmen der Absicherung sehr alter Betriebssysteme wie beispielsweise Windows 95/98/NT wird hier verzichtet, da diese Systeme seit geraumer Zeit nicht mehr unterstützt werden und somit auch keine Aktualisierungen zu ihrer Absicherung (*Security Patches*) erhalten.

Virtual Private Network (VPN)

Das Internet ist als weltweit umspannendes Netzwerk zur Übertragung allgemein öffentlicher Daten mittlerweile überall anerkannt. Ob es darum geht, sich über das *World Wide Web* Informationen zu beschaffen, Online-Bestellungen aufzugeben oder per *E-Mail* Nachrichten zu versenden: Das Internet bietet ein geeignetes Instrumentarium für eine globale Kommunikation.

Die Akzeptanz schwindet allerdings, sobald es im privaten wie im geschäftlichen Umfeld um die Übertragung vertraulicher Informationen geht. Hier werden in der Regel dedizierte Kommunikationsverbindungen verwendet, die beide Kommunikationspartner direkt miteinander verbinden. Alternativ werden von vertraulichen Providern Kommunikationsleitungen angemietet und ein eigenes Unternehmensnetzwerk aufgebaut. Als problematisch erweisen sich hier allerdings die äußerst hohen Kosten, die bei der Implementierung komplexer Weitverkehrsnetzwerke entstehen.

Eine durchaus interessante Alternative stellt hier der Gedanke einer »gesicherten Datenübertragung über das Internet« dar. Grundsätzlich ist die Kommunikation über das Internet natürlich als unsicher zu bezeichnen und muss daher für die Übertragung sensibler Unternehmensdaten abgelehnt werden. Könnte man allerdings die Kommunikation gegenüber Fremden absichern und abhörsicher gestalten, so würde diese Alternative wieder neu diskutiert. In dem Zusammenhang stellt die Nutzung sogenannter virtueller privater Netzwerke (VPN = *Virtual Private Network*) eine sehr interessante Alternative dar. Aus [Abbildung 7-7](#) ist ersichtlich, dass effektive VPNs einen verschlüsselten »Datentunnel« zwischen zwei Kommunikationspartnern bilden können. Natürlich können

auch »einfache« Router für den Aufbau von VPNs verwendet werden.

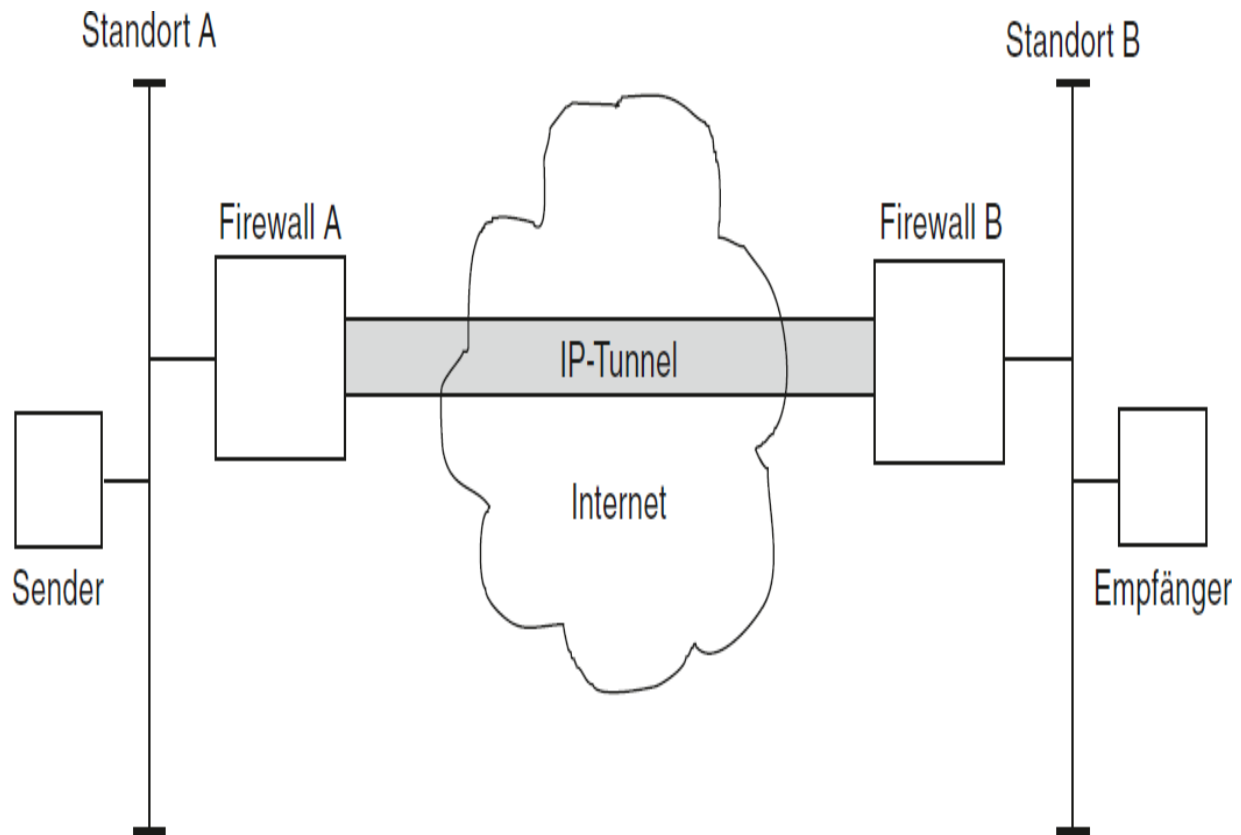


Abb. 7–7 Prinzip virtueller privater Netzwerke über Firewalls

Bei einer VPN- bzw. SSL-VPN-Verbindung (SSL = *Secure Sockets Layer*) werden zwischen Systemen der beiden Kommunikationsendpunkte Verfahren zur Schlüsselgenerierung, zum Schlüsselaustausch und zur Verschlüsselung ausgehandelt, die dann während des gesamten Kommunikationsverlaufs Gültigkeit besitzen.

Es gibt verschiedene konzeptionelle Ansätze zur Bildung von VPNs. Sie können auf verschiedenen Schichten des ISO/OSI-Referenzmodells implementiert (meist Schicht 2 und 3) und in unterschiedlichen Kommunikationsbeziehungen eingerichtet werden. In den meisten Fällen werden zwei separate Netzwerke über ein VPN der Netzwerkschicht (IP) miteinander verbunden. Dies wird entweder über Router oder auch über Firewall-Systeme realisiert.

Es sind allerdings auch Situationen denkbar, in denen ein sogenannter *Client-VPN* etabliert werden muss. Dabei handelt es sich um eine verschlüsselte Kommunikation zwischen einem einzelnen Client-Rechner und einem entfernten Serversystem. Verbindungen dieser Art werden normalerweise über RAS-Dienste (RAS = *Remote Access Service*) eingerichtet, bei denen Wählverbindungen benutzt werden. Dies kann beispielsweise über bereits implementierte Konfigurationsoptionen in den Netzwerkumgebungen der Betriebssysteme erfolgen oder durch besondere Wählverbindungen zu Firewall-Systemen.

Gründe für den Einsatz einer VPN-Verbindung sind beispielsweise:

- Absicherung interner Datenströme im eigenen Unternehmensnetzwerk (z.B. sensible Personaldaten)

- Beim »Durchqueren« öffentlicher Netzwerke (z.B. Internet) soll der unternehmensinterne Datenstrom gegenüber Dritten geschützt sein.

Sicherheitsprotokoll IPsec

Ein nicht unerhebliches Problem des Internet Protocol (IP) stellt das Fehlen wesentlicher Sicherheitsmerkmale und -funktionen dar. Dieses zu den ältesten Standards (J. Postel: Internet Protocol, RFC 791, September 1981) gehörende Protokoll weist einige gravierende Sicherheitsdefizite auf wie beispielsweise fehlende Datenintegrität, Authentifizierungsmechanismen und unzureichende oder fehlende Vertraulichkeit.

Diese Probleme sind beim unter RFC 2401 vom November 1998 eingeführten Protokoll *IPsec (IP Security)* und seiner Aktualisierung in RFC 4301 vom Dezember 2005 weitgehend gelöst. Das »Mithören« von Daten wird durch eine geeignete Datenverschlüsselung verhindert, die Gefahr einer illegalen Veränderung der Originaldaten durch einen »elektronischen Fingerabdruck« ausgeschlossen und die Sicherheit, dass es sich beim Absender tatsächlich um die angegebene Instanz handelt, durch einen privaten Schlüssel bzw. die digitale Signatur gewährleistet.

IPsec-Merkmale

Die Charakteristik des IPsec wird grundsätzlich in zwei Teilbereichen beschrieben:

- Symmetrische Verschlüsselung (ESP = *Encapsulated Security Payload*) der Nutzdaten und Nachrichtenauthentifizierung (AH = *Authentication Header*) von IP-Datenpaketen innerhalb des *IP Security Protocol* (IPSP)
- Schlüsselmanagement innerhalb des *Internet Key Management Protocol* (IKMP)

ESP kann in zwei unterschiedlichen Modi betrieben werden. Während der Transportmodus lediglich die Nutzdaten verschlüsselt, wird im Tunnelmodus das gesamte IP-Paket einschließlich IP-Header verschlüsselt und in ein weiteres, für den Transport notwendiges Paket eingefügt. Somit ist das originäre Datenpaket während des gesamten Transports über das Netzwerk nicht lesbar. Dieses Verfahren wird primär beim Betrieb von *Virtual Private Networks* (VPN) über das Internet eingesetzt.

IKMP verwendet heute primär das Protokoll *Oakley* für den Schlüsselaustausch der Kommunikationspartner und das Protokoll *ISAKMP* für die Verwaltung der *Security Associations* (SA; werden zur Herstellung einer kryptografisch relevanten

Kommunikationsbeziehung zwischen zwei oder mehreren Partnern benötigt).

Das als weniger aufwendig, aber auch weniger sicher geltende *Simple Key Management Protocol* (SKIP) hat sich bislang nicht durchsetzen können, wurde allerdings in zahlreichen Implementierungen realisiert. Im Gegensatz zu ISAKMP, das auf dem ISO/OSI-Layer 7 operiert, bleiben sämtliche SKIP-Aktivitäten auf der Netzwerkschicht, dem Layer 3.

Bei den Merkmalen einer IPsec-Verbindung sind insbesondere folgende Merkmale zu berücksichtigen bzw. von Relevanz:

- *Datenintegrität*

Während der Datenübertragung muss sichergestellt sein, dass eine Veränderung der Daten zwischen Sender und Empfänger nicht möglich ist. Dies wird über die Verwendung privater Schlüssel und Hash-Werte realisiert, deren Auflösung lediglich vom Sender und vom Empfänger vorgenommen werden kann.

- *Authentifizierung*

Die Sicherheitsmechanismen (der private Schlüssel als digitale Signatur) zur Wahrung der Datenintegrität liefern

darüber hinaus auch den Beweis, dass die vom Sender verschickten Datenpakete auch tatsächlich vom angenommenen Sender stammen und nicht »unterwegs« durch einen Dritten verändert wurden.

- *Vertraulichkeit*

Eine Verschlüsselung der Daten sorgt dafür, dass sie Dritten nicht zugänglich gemacht werden können. Damit dies sichergestellt ist, können ständig wechselnde Schlüssel nach bestimmten Verfahren (z.B. dem *Internet Key Exchange* = IKE) eingesetzt werden. Die infrage kommenden Betriebsmodi sind der *Transportmodus* (Ende-zu-Ende-Verschlüsselung) und der *Tunnelmodus* (geografische Daten, wie z.B. die Sende- und Empfangsadresse, werden im Datenteil eines neu generierten IPsec-Pakets integriert).

- *Schutzmechanismen*

Zur Vermeidung direkter Angriffe auf Datenpakete werden besondere Sicherheitsfunktionen zur Verfügung gestellt. So führt beispielsweise der Anti-Replay-Service (ARS) Prüfungen durch, ob ein aktuell erhaltenes Datenpaket schon einmal empfangen wurde. Zahlreiche Wiederholungen dieser Art, die illegal von Dritten durchgeführt werden, um Zielsysteme zu blockieren oder zu »verwirren«, können somit analysiert und verhindert werden. Darüber hinaus lassen sich »Denial-of-Service«-Attacken (*Denial of Service Attack* = Angriffe, bei

denen die Ressourcen wichtiger Serversysteme durch Überlastung blockiert und somit unbrauchbar werden) dadurch wirkungsvoll verhindern, dass die Sicherheits-Gateways (z.B. Firewalls oder auch dafür explizit konfigurierte Router) nur noch bestimmte Tunnelprotokolle (z.B. IPSec ESP, IPSec AH) zulassen und somit Angriffe auf anderen Protokollebenen (z.B. TCP) überhaupt nicht mehr stattfinden können, da es sie nicht mehr gibt.

IP- und IPsec-Paketformat

IP und IPsec weisen unterschiedliche Paketformate auf. Für das Internet Protocol (IP) sieht das Paketformat so wie in [Abbildung 7-8](#) dargestellt aus.

Version	Header Length	Type of Service	Packet Length	
Identification			Flags	Fragment Offset
Time to Live		Protocol	Header Checksum	
Source IP Address				
Destination IP Address				
Options/Padding				

Abb. 7–8 *IP-Protokoll-Header*

IPsec wird dahin gehend erweitert, dass der *IP-Protokoll-Header* um Elemente zur Verschlüsselung und Authentifizierung ergänzt wird. Wie bereits dargestellt, wird dabei jeweils zwischen dem Transport- und Tunnelmodus innerhalb der beiden Protokollvarianten ESP (*Encapsulated Security Payload*) und AH (*Authentication Header*) unterschieden.

Die [Abbildung 7–9](#) bis [Abbildung 7–12](#) zeigen den grundsätzlichen Aufbau der Paketformate. Hier wird deutlich, dass im Unterschied zu einem »normalen« IP-Datenpaket ein Teil des IPsec-Datenpakets authentifiziert und verschlüsselt

wird. Beim AH-Protokoll hingegen fehlt die Verschlüsselung. Allerdings wird für die Authentifizierung auch der reale IP-Header berücksichtigt. Dies bewirkt eine Erhöhung der Authentifizierungsqualität, da ja das gesamte Paket authentifiziert wird.

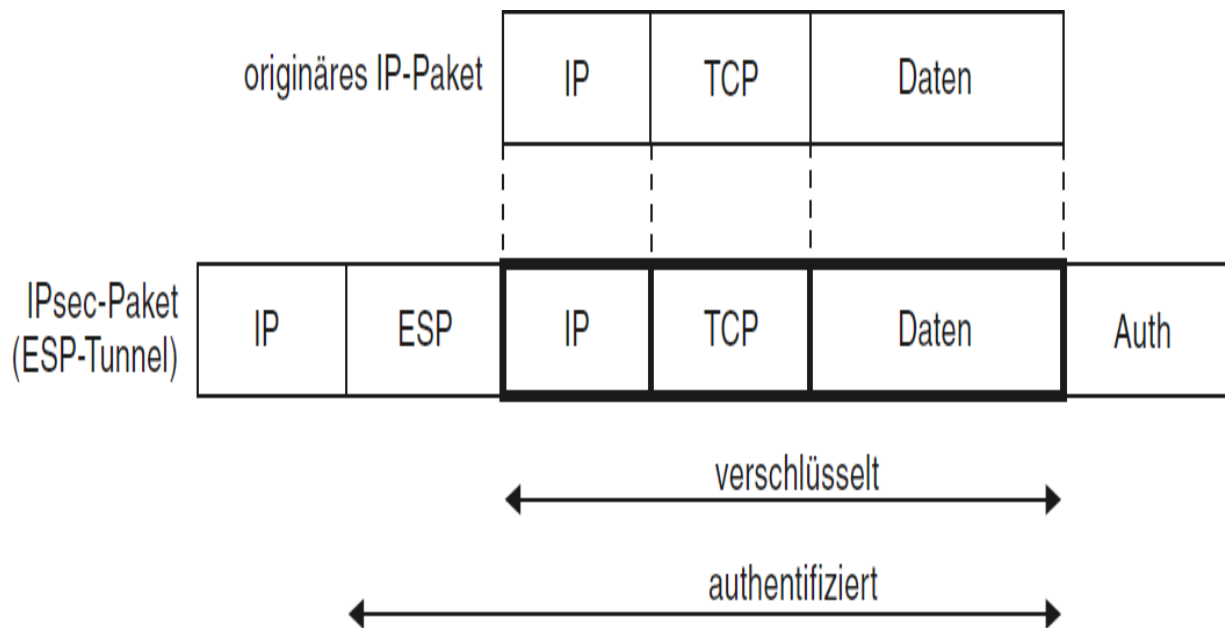


Abb. 7–9 IPsec-Paketformat für das ESP-Protokoll im Tunnelmodus

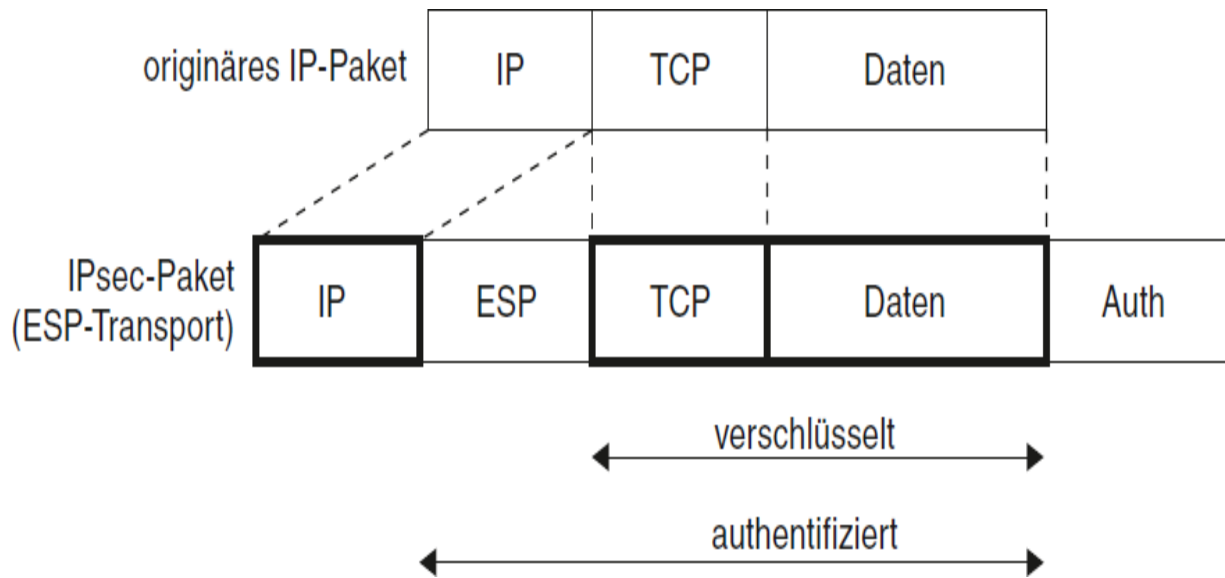


Abb. 7–10 IPsec-Paketformat für das ESP-Protokoll im Transportmodus

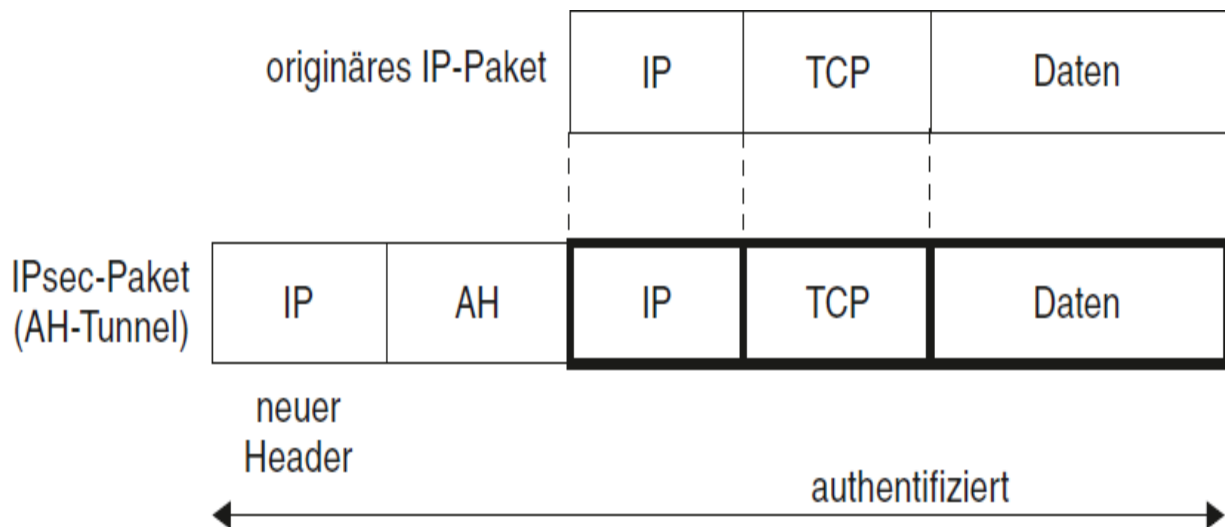


Abb. 7–11 IPsec-Paketformat für das AH-Protokoll im Tunnelmodus

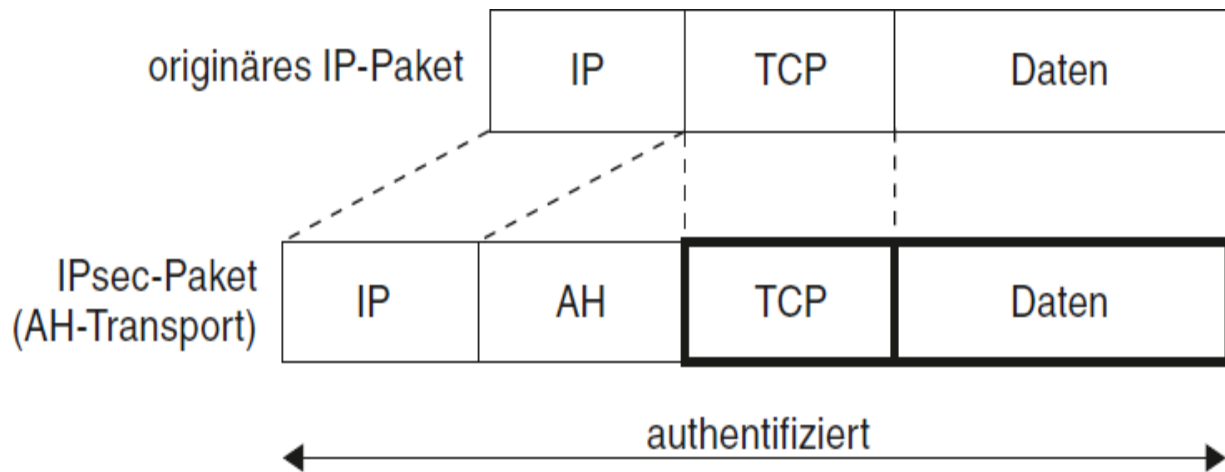


Abb. 7–12 IPsec-Paketformat für das AH-Protokoll im Transportmodus

Transport- und Tunnelmodus

IPsec-Datenpakete lassen sich grundsätzlich unter Verwendung des Transportoder des Tunnelmodus übertragen. Wo die Unterschiede liegen, wird nachfolgend erläutert.

Transportmodus

Der Transportmodus umfasst eine Ende-zu-Ende-Kommunikation, bei der lediglich die Verschlüsselung des Datenteils eines IP-Pakets erfolgt, so wie in [Abbildung 7–13](#) beispielhaft dargestellt.

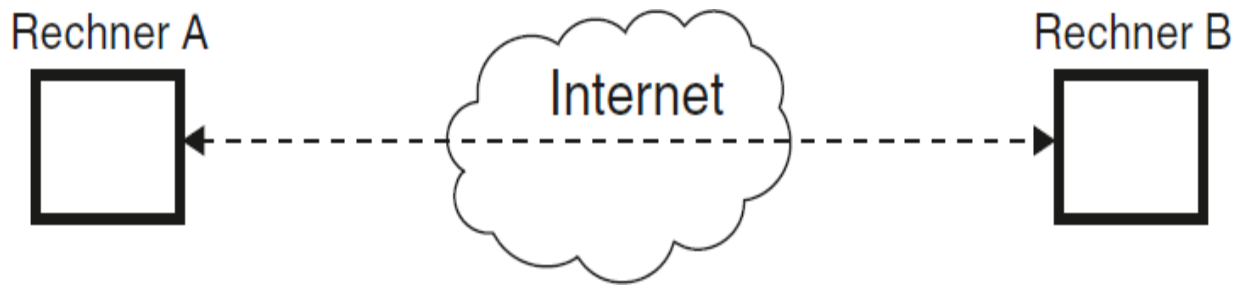


Abb. 7–13 *Transportmodus*

Dieses Verfahren (Transportmodus) wird primär zum Schutz der höheren Protokollschichten (ab TCP/UDP) eingesetzt. Der *IP-Header* – und somit auch Quell- und Ziel-IP-Adresse – bleibt unverändert.

HINWEIS

Der Transportmodus kann sowohl für *Encapsulating Security Payload* (ESP) als auch – sofern lediglich Anforderungen zur Authentifizierung und nicht zur Verschlüsselung gestellt sind – für *Authentication Header* (AH) eingesetzt werden.

Tunnelmodus

Beim Tunnelmodus wird auch der *IP-Header* verschlüsselt, sodass die Generierung eines neuen *IP-Headers* erforderlich wird, der das Routing des Pakets vornehmen muss. Dieses Verfahren gewährleistet eine sehr hohe Datensicherheit, da das gesamte Originalpaket unkenntlich gemacht wird und somit für Dritte nicht mehr ausgelesen werden kann. Der Tunnelmodus

(siehe [Abb. 7-14](#)) kommt meist dann zum Einsatz, wenn aus den jeweiligen Zielnetzen ein geringeres Gefährdungspotenzial hervorgeht (meist handelt es sich um interne Netze) als vom eigentlichen Transportnetz (Internet).

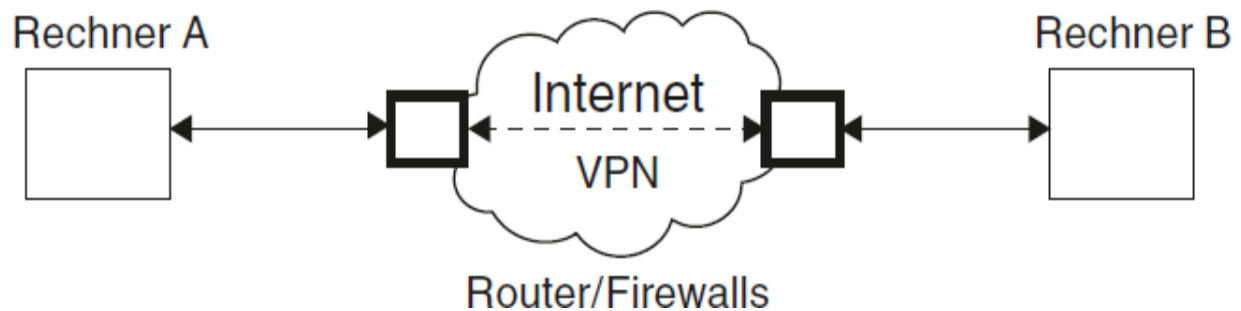


Abb. 7-14 Tunnelmodus

IPsec-Protokolle AH und ESP

Beim *Authentication Header Protocol* (AH) und bei dem *Encapsulating Security Payload Protocol* (ESP) handelt es sich um die zentralen IPsec-Protokolle. Beim AH erhält das Protokollfeld im IP-Header den Wert 51, beim ESP ist der Wert 50 eingetragen.

- *Authentication Header Protocol (AH)*

Das *Authentication Header Protocol* ist primär für die Sicherung der Datenintegrität und Authentifizierung der Datenpakete zuständig. Optional bietet es Schutz vor wiederholtem Senden von Datagrammen (Replay-Angriffe), indem für die Datenpakete fortlaufende Sequenznummern

generiert und vom Empfänger ausgewertet werden. Da das gesamte IP-Datenpaket gesichert wird (auch der IP-Header), ist die Verwendung von *Network Address Translation* (NAT) nicht möglich, denn die bei NAT übliche Manipulation von IP-Headern würde die AH-Datenintegrität verletzen. Der Aufbau eines IPsec-AH-Pakets ist [Abbildung 7-15](#) zu entnehmen:

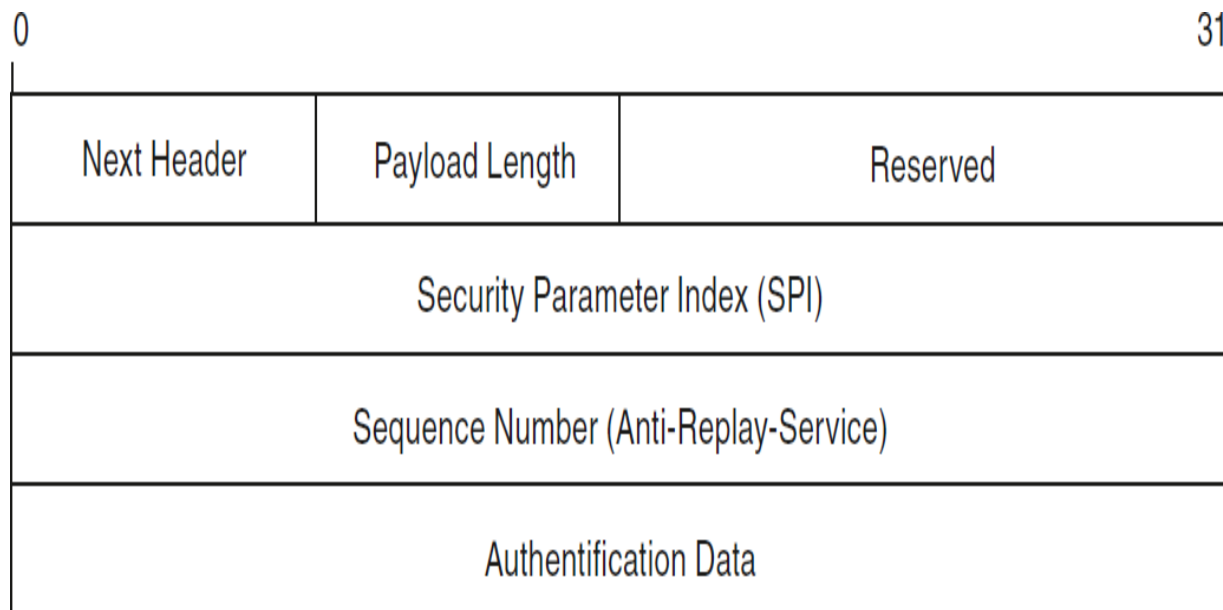


Abb. 7-15 *IPsec-AH-Paketformat*

- *Next Header* (8)

Dieses Feld gibt den Datentyp an, der dem Authentication Header folgt. Die Angabe erfolgt als IP-Protokollnummer.

- *Payload Length* (8)

Dieses Feld gibt die Länge des Authentication Header in 4-Byte-Wörtern (32 Bit) an.

- *Reserved* (16)

Dieses Feld wird nicht benutzt und besitzt den Wert 0.

- *Security Parameter Index* (32)

Dieses Feld beinhaltet einen eindeutigen 32-Bit-Wert, der gemeinsam mit der IP-Zieladresse und dem Sicherheitsprotokoll (AH) die für dieses Datenpaket relevante *Security Association* (SA) identifiziert.

- *Sequence Number* (32)

Dieses Feld enthält einen stetig steigenden 32-Bit-Zähler, der bei jedem neuen Datenpaket um 1 erhöht wird. Er wird – sofern konfiguriert – dazu eingesetzt, das wiederholte Senden von Datenpaketen zu vermeiden. Dieses Feld wird vom Empfänger des Datenpakets ausgewertet.

- *Authentication Data* (variabel)

In diesem Feld werden die eigentlichen Authentifizierungsdaten untergebracht. Der *Integrity Check Value* (ICV) wird nach bestimmten Verfahren ermittelt, die im *Internet Key Exchange Protocol* (IKE) festgelegt werden.

- *Encapsulating Security Payload Protocol* (ESP)

Das *Encapsulating Security Payload Protocol* liefert zusätzlich zu den vom Authentication Header Protocol zur Verfügung gestellten Mechanismen den Schutz der Datenvertraulichkeit. Die Besonderheit gegenüber dem AH-

Protokoll besteht darin, dass ein Teil des Datenpakets verschlüsselt wird.

Im Transportmodus erstreckt sich die Verschlüsselung über den *TCP-Header* und den Datenteil. Im Tunnelmodus wird der originäre *IP-Header* ebenfalls verschlüsselt und um einen neuen *IP-Header* erweitert, der dann die für das Routing relevanten Informationen enthält. Der verschlüsselte Teil wird von einem ESP-Header (folgt unmittelbar nach dem IP-Header) und einem AUTH-Header (schließt das Datenpaket ab) eingeschlossen (siehe [Abb. 7-16](#)).

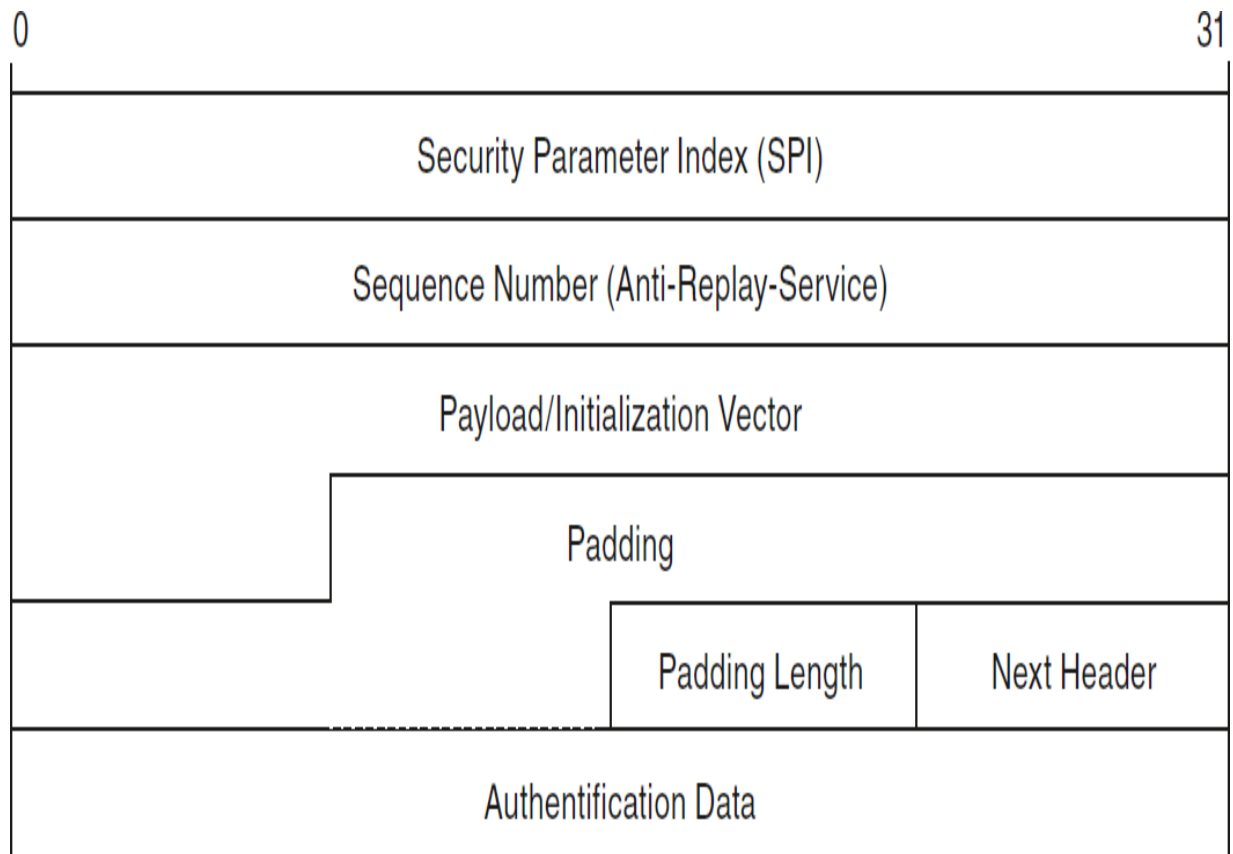


Abb. 7–16 *IPsec-ESP-Datenformat*

- *Security Parameter Index (8)*

Dieses Feld beinhaltet einen eindeutigen 32-Bit-Wert, der gemeinsam mit der IP-Zieladresse und dem Sicherheitsprotokoll (ESP) die für dieses Datenpaket relevante Security Association (SA) identifiziert.

- *Sequence Number (32)*

Siehe Erläuterungen zu *Sequence Number* beim AH-Paketformat im vorigen Abschnitt.

- *Payload (variabel)*

Dieses Feld enthält die vom Next-Header-Feld beschriebenen Nutzdaten. Sollte der verwendete Verschlüsselungsalgorithmus Synchronisationsdaten erfordern (z.B. einen Initialisierungsvektor), so können diese Informationen hier ebenfalls untergebracht werden.

- *Padding* (0-255)

Die Länge dieses Felds wird durch Anforderungen bei der Verwendung bestimmter Verschlüsselungsalgorithmen bestimmt, die eine konkrete Blockgröße benötigen (bei DES beispielsweise 64 Bit). Dieses Feld sorgt dann für ein entsprechendes »Auffüllen« und für die korrekte Positionierung der beiden folgenden Felder *Padding Length* und *Next Header*.

- *Padding Length*

Dieses Feld gibt die Anzahl der Bytes des *Padding*-Felds an. Der Wert 0 besagt, dass es kein *Padding*-Feld gibt.

- *Next Header* (8)

Der Wert in diesem Feld gibt den Datentyp im *Payload*-Feld an. Er lautet beispielsweise 4 (IP), 6 (TCP) oder 17 (UDP).

- *Authentication Data* (variabel)

Siehe Erläuterungen zu *Authentication Data* beim AH-Paketformat im vorigen Abschnitt – allerdings entfällt dieses Feld, wenn keine Authentifizierung vorgenommen wurde (Authentifizierung ist bei ESP optional).

Internet Key Exchange (IKE)

Während IPsec das Problem des sicheren Schlüsselaustauschs grundsätzlich vernachlässigt bzw. diesen manuell vornimmt, beschreibt IKE (*Internet Key Exchange*) eine Fülle von Mechanismen, die den Prozess des Schlüsselaustauschs *dynamisch* durchführen.

Will man auf einer Kommunikationsverbindung zweier Partner die für eine Verschlüsselung erforderlichen Schlüssel übertragen, so muss sichergestellt sein, dass die Schlüssel während der Übermittlung nicht abgefangen und identifiziert werden können. Folgende Protokolle bzw. Mechanismen sorgen für einen sicheren Austausch von Schlüsseln:

- Internet Security Association and Key Management Protocol (ISAKMP)
- ISAKMP Domain of Interpretation (DoI)
- OAKLEY Key Determination Protocol
- IKE-Modi

ISAKMP

Das *Internet Security Association and Key Management Protocol* (ISAKMP) definiert Prozeduren und Paketformate zur Erstellung, Verhandlung, Modifikation und Löschung

sogenannter *Security Associations* (SA). SAs beinhalten alle Informationen, die zur Ausführung verschiedener Netzwerksicherheitsdienste erforderlich sind (z.B. AH oder ESP).

ISAKMP definiert Nutzdaten (*Payloads*) zum Schlüsselaustausch und zur Authentifizierung, deren Formate das Gerüst für den Schlüsseltransport und die Übertragung von Authentifizierungsdaten liefern. Diese sind unabhängig von der eigentlichen Technik zur Schlüsselgenerierung, dem Verschlüsselungsalgorithmus und dem Authentifizierungsmechanismus. ISAKMP konzentriert sich vielmehr auf das Management der Security Associations und sorgt somit dafür, dass sich andere Sicherheitsprotokolle mit der Verwaltung von SAs nicht mehr beschäftigen müssen.

HINWEIS

Die aus Sicherheitsgründen beste Methode zur Übertragung von Authentifizierungsdaten ist der Einsatz digitaler Zertifikate und damit die Verwendung öffentlicher und privater Schlüssel. Dabei ist die ISAKMP-Kommunikation als besonders kritisch anzusehen, da sie die Grundlage für die Übertragung aller weiteren sicherheitsrelevanten Daten darstellt. Werden bereits in diesem frühen Stadium der Kommunikation Informationen

abgefangen, analysiert und entschlüsselt bzw. missbraucht, so besteht die Gefahr, dass die gesamte Folgekommunikation durch Dritte manipuliert oder ausspioniert werden und somit einem Unternehmen großer Schaden entstehen kann.

Das ISAKMP-Paket besteht aus einem Protokoll-Header mit fester Größe und einem variablen Teil mit ISAKMP-Nutzdaten (*Payload*). Grundsätzlich erfolgt die Durchführung der Austauschaktivitäten im ISAKMP in zwei Phasen:

- Phase 1:
Aufbau einer ISAKMP-SA
- Phase 2:
Austauschvorgänge der Sicherheitsprotokolle

Folgende Austauschverfahren sind dabei für die Phase 2 definiert:

- Base Exchange
- Identity Protection Exchange
- Authentication Only Exchange
- Aggressive Exchange
- Informational Exchange

Innerhalb von ISAKMP wird eine *Domain of Interpretation* (DoI) eingesetzt, um Protokolle, die ISAKMP zur Aushandlung von

Security Associations verwenden, zu gruppieren. ISAKMP stellt folgende Anforderungen an eine DoI:

- Definition eines Namensschemas für DoI-spezifische Protokollidentifikatoren
- Interpretation des *Situation Field*
- Definition der anwendbaren Sicherheitsrichtlinien (*Security Policies*)
- Definition der Syntax für DoI-spezifische SA-Attribute in Phase 2
- Definition der Syntax für DoI-spezifischen *Payload*-Inhalt
- Definition zusätzlicher Schlüsselaustauschtypen, sofern benötigt
- Definition zusätzlicher Benachrichtigungstypen (*Notification Message Types*), sofern benötigt

HINWEIS

Sämtliche *Domains of Interpretation* müssen beim IANA (*Internet Assigned Numbers Authority*) registriert werden.

OAKLEY Key Determination Protocol

Das *OAKLEY Key Determination Protocol* ist ein Schlüsselaustauschprotokoll, das auf dem Diffie-Hellman-Algorithmus basiert und somit für zwei authentifizierte

Kommunikationspartner einen gesicherten Austausch von Schlüsseln durchführen kann.

Während ISAKMP grundsätzlich die Rahmenbedingungen vorgibt, stellt OAKLEY die eigentliche Ausführung der Austauschvorgänge in den beiden Phasen dar. Die Umgebung, in der ein Schlüsselaustausch stattfindet, wird »Gruppe« genannt, wobei standardmäßig sechs Gruppen definiert sind:

- Gruppe 0:
nicht definiert
- Gruppe 1:
Modulare Exponentiation unter Verwendung einer 768 Bits großen Primzahl
- Gruppe 2:
Modulare Exponentiation unter Verwendung einer 1024 Bits großen Primzahl
- Gruppe 3:
Modulare Exponentiation unter Verwendung einer 1536 Bits großen Primzahl
- Gruppe 4:
Elliptische Kurve mit 2155 Feldelementen
- Gruppe 5:
Elliptische Kurve mit 2185 Feldelementen

HINWEIS

Die Implementierung der Gruppe 1 im IKE-Protokoll ist obligatorisch. Alle anderen Gruppen sind optional.

IKE-Modi

Zur Durchführung der Austauschmechanismen innerhalb des IKE-Protokolls sind gemäß Standardvorgabe die folgenden Modi definiert:

- *IKE-Main-Mode*

Die ersten beiden Nachrichten handeln die *Policy* aus und die nächsten beiden sorgen für den Austausch der öffentlichen Diffie-Hellman-Werte und weiterer Daten. Die letzten beiden Nachrichten führen die Authentifizierung des Diffie-Hellman-Austauschs durch.

- *IKE-Aggressive-Mode*

Die ersten beiden Nachrichten handeln die *Policy* aus und sorgen für den Austausch der öffentlichen Diffie-Hellman-Werte und untergeordneter, aber ebenfalls erforderlicher Daten sowie der Identitäten. Die zweite Nachricht übernimmt zusätzlich die Authentifizierung des Empfängers (*Responder*). Die dritte Nachricht authentifiziert den Sender

(*Initiator*) und liefert einen Beweis der Teilnahme am Austauschvorgang.

- *IKE-Quick-Mode*

Dieser Modus stellt keinen eigenständigen Austauschvorgang dar, wird jedoch als Teil des SA-Aushandlungsprozesses (Phase 2) verwendet, um Verschlüsselungsmaterial zu gewinnen und die gemeinsame *Policy* für nicht zum ISAKMP gehörende Security Associations auszuhandeln.

- *New Group Mode*

Während der Main Mode und der Aggressive Mode in der Phase 1 für den authentifizierten Schlüsselaustausch verantwortlich sind, wird der Quick Mode ausschließlich in Phase 2 tätig.

Eine der wichtigsten IKE-Optionen ist die *Perfect Forward Secrecy* (PFS). Wird diese Option nicht verwendet, so werden alle innerhalb einer Schlüsselgenerierungssequenz generierten Schlüssel für eine *Security Association* aus denselben Grunddaten erzeugt. Wird also ein Schlüssel von Hackern identifiziert und geknackt, so können weitere später generierte Schlüssel daraus rekonstruiert und damit verschlüsselte Daten entschlüsselt werden. Bei aktiviertem PFS erfolgt eine Schlüsselgenerierung stets auf der Basis neuen Grundmaterials, sodass eine einzige Schlüsselkompromittierung weitere Schlüssel in ihrer Geheimhaltung nicht beeinträchtigt. Eine

Verringerung der Zeitintervalle, in denen ein völlig neuer Schlüssel generiert wird, sorgt für zusätzliche Sicherheit, da die Gültigkeitsdauer des jeweiligen Schlüssels verkürzt wird.

In der SA-Aushandlung legt IKE fest, welches Authentifizierungsprinzip zur Anwendung kommen soll. Es gibt vier verschiedene Möglichkeiten:

- *Pre-Shared Key*

Die vereinbarte Hash-Funktion wird auf den Hauptschlüssel (berechnet aus dem *Pre-Shared Key*), die SA-Nutzdaten, die öffentlichen Diffie-Hellman-Werte, die beiden Cookies und die Identitäten von Initiator und Responder angewendet.

- *Digitale Signatur*

Die Signatur wird unter Verwendung eines privaten Schlüssels und eines Signaturalgorithmus (z.B. RSA) vorgenommen. Optional ist auch eine Überprüfung durch Zertifikate und die dort bereitgestellten öffentlichen Schlüssel möglich. Die Zertifikate werden dann gemeinsam mit der Signatur übertragen.

- *Public-Key-Verschlüsselung (RSA)*

Diese sehr rechenintensive Authentifizierungsmethode (erhöhte Public-Key-Verschlüsselungen und -Entschlüsselungen auf jeder Seite der Kommunikation)

erfordert normalerweise wesentlich leistungsfähigere Hardware und wird daher nur sehr selten eingesetzt.

- *Revised Mode der Public-Key-Verschlüsselung*

Hierbei handelt es sich um eine vereinfachte Public-Key-Verschlüsselung, bei der die Anzahl von Ver- und Entschlüsselungen deutlich reduziert wird.

HINWEIS

In der Praxis werden zur Authentifizierung und aus Performancegründen primär der *Pre-Shared Key* und die *Digitale Signatur* verwendet, da beim Einsatz der Public-Key-Verschlüsselung oft kostspielige Hardwarebeschleuniger (spezielle Controller für Router oder PC) erforderlich werden.

Weiterführende Informationen zu IPsec sind beispielhaft in den einschlägigen RFCs nachzulesen.

Weitere Überlegungen

Zum Thema IT-Sicherheit gibt es natürlich weitere, nicht unmittelbar mit der TCP/IP-Protokollfamilie in Zusammenhang stehende Themen, auf deren Vorstellung im Rahmen dieses Buches jedoch verzichtet wird. Die im Folgenden kurz dargestellten Bereiche bilden allerdings

Sicherheitsschwerpunkte, auf die dennoch hingewiesen werden soll.

Grundschutzhandbuch für IT-Sicherheit des BSI

Das Grundschutzhandbuch für IT-Sicherheit des BSI beschreibt Standardsicherheitsfragen, die im heutigen Umgang mit IT-Systemen und den zu schützenden Unternehmensressourcen relevant sind. Es werden grundsätzliche Maßnahmen vorgeschlagen, die einen Basisschutz für diese Ressourcen gewährleisten sollen. Durch eine derzeit halbjährliche Aktualisierung des Handbuches ist eine sehr realitätsnahe Diskussion der einzelnen Themen gewährleistet.

Das Grundschutzhandbuch kann sowohl für den Einstieg in die Sicherheitsdiskussion als auch für die Weiterentwicklung einer bereits vorhandenen Sicherheitspolitik sehr hilfreich sein und wertvolle Informationen liefern.

Patching

Unter dem englischen auch im deutschen Sprachgebrauch üblichen Begriff »Patching« versteht man die Aktualisierung von Betriebssystemen mit Sicherheitsfunktionen. Diese werden von den Herstellern der Betriebssysteme regelmäßig öffentlich zur Verfügung gestellt (z.B. durch Downloads) oder sind

Bestandteil von Serviceleistungen beauftragter Provider bzw. Sicherheitspartner von Unternehmen. Grundsätzlich sind zwei unterschiedliche Ansätze für ein effektives Patching möglich:

- *Eigene Infrastruktur* – dabei entwickelt ein Unternehmen im eigenen Netzwerk ein aus mehreren Serversystemen bestehendes Patching, dem eine eigens entwickelte Patching-Strategie zugrunde liegt (beispielsweise ein aus mehreren Stufen bestehender Prozess für Server mit abgestufter Relevanz von Unternehmensdaten).
- *Patching als externer Service* – hier wird das gesamte Patching an einen externen Dienstleister ausgelagert, der i.d.R. eigene Komponenten in das Unternehmensnetzwerk installiert; heute oft auch als Cloud-Service.

Die durch zahlreiche Angriffe auf die verwundbaren Systeme eines Unternehmens verursachten zum Teil erheblichen Schäden haben dazu geführt, dass effektives Patching mittlerweile als einer der wichtigsten Sicherheits-Services im Unternehmen angesehen wird. Im jährlichen IT-Budget nimmt dieses Thema eine wesentliche Rolle ein.

Bei allem Engagement für die notwendigen Sicherheits-Updates sind gründliche Testläufe mit aktualisierten Betriebssystemen in eigens dafür vorgesehenen Testumgebungen unabdingbar.

Wird darauf aus Kostengründen verzichtet, können plötzlich auftretende Inkompatibilitäten der neuen mit zusätzlichen Sicherheitsfunktionen versehenen Betriebssysteme zu vermeidbaren Rechner-Abstürzen oder zu schwerer identifizierbaren Problemen führen. Ein »Test before implement« (Testen vor Einsatz) lohnt sich immer.

Der Schutz des Perimeters

Zur Absicherung eines Unternehmensnetzwerks und seiner Ressourcen nach außen gibt es unterschiedliche Ansätze. In jeder der denkbaren Konfigurationen steht allerdings ein Firewall-System als zentrale Sicherheitskomponente im Mittelpunkt. Je nach Firewall-Typ sind ergänzende Systeme erforderlich, um ein Grundschutzbedürfnis befriedigen zu können. Unabhängig von individuellen Anforderungen kann dieser Schutz durch die Implementierung von drei unterschiedlichen Sicherheitskomponenten gewährleistet werden:

- Firewall-System
- Content Security System
- Intrusion Detection/Protection System

Hauptaufgabe des *Firewall-Systems* ist die Realisierung eines Zugriffsschutzes gegenüber nicht autorisierten Fremdzugriffen

auf das eigene Netzwerk und seine Ressourcen. Je nach Produktauswahl werden auch weitere Funktionen zur Verfügung gestellt (z.B. Betrieb von virtuellen privaten Netzwerken, Network Address Translation oder auch Überprüfung von Dateninhalten in Zusammenarbeit mit anderen Softwareprodukten).

Ein *Content Security System* ist für die Überprüfung von Dateninhalten zuständig. Ein klassisches Beispiel stellt hier ein zentraler Virens Scanner dar, der bereits an der Firewall den eingehenden Datenverkehr auf Viren oder andere Objekte mit entsprechendem Gefährdungspotenzial überprüft und somit ihr Eindringen ins interne Netzwerk verhindern kann. Dieser Server verrichtet seine Dienste innerhalb eines ausgelagerten Netzwerksegments, das *Demilitarized Zone* (DMZ) genannt wird.

IDS/IRS-Systeme (*Intrusion Detection* bzw. *Intrusion Response Systems*) stellen gewissermaßen die »Netzwerk-Polizei« dar und können den gesamten Datenverkehr innerhalb eines Netzwerksegments oder auch auf einem einzelnen Serversystem auf mögliche Angriffe von Crackern überprüfen und im Zusammenspiel mit Firewall-Systemen abwehren.

Public Key Infrastructure (PKI)

Eine exakte Definition des Begriffs PKI lässt sich in der Fachliteratur nur schwer finden. Überall dort, wo über dieses Thema geschrieben oder gesprochen wird, erfolgt eine mehr oder weniger intensive Diskussion über konkrete Teilaspekte mit unterschiedlicher bedarfsorientierter Gewichtung. Zu »PKI« gehört eine recht umfangreiche Liste sicherheitsrelevanter Themen wie beispielsweise:

- Authentifizierung
- Verschlüsselung
- Zertifikate
- Digitale Unterschrift

Diese keineswegs vollständige Übersicht umfasst zumindest die wesentlichen Aspekte der *Public Key Infrastructure*.

Security Incident und Event Management (SIEM)

Wie die deutsche Übersetzung von SIEM nahelegt, bedeutet die Einführung von SIEM in einem Unternehmen das Registrieren und Darstellen sämtlicher Ereignisse aller in einem Netzwerk angebundenen Systeme. Diese Informationen werden in Echtzeit transparent dargestellt und ermöglichen eine weitgehend verzögerungsfreie Reaktion, einschließlich der

Durchführung geeigneter Gegenmaßnahmen zu beobachteten Sicherheitsvorfällen.

SIEM ist die Grundlage eines jeden Security Reporting und ist oft Bestandteil entsprechender Sicherheits-Audits. Auf dem Markt verfügbare Software (beispielsweise SPLUNK; siehe hierzu auch www.splunk.com) übernimmt Monitoring, Reporting und Administration der relevanten Sicherheitsvorfälle bzw. verdächtiger Symptome.

Datenschutz-Grundverordnung (DSGVO)

Am 25. Mai 2018 wurde die Datenschutz-Grundverordnung innerhalb der Europäischen Union verabschiedet. Im Artikel 1 der DSGVO sind die drei zentralen Ziele beschrieben:

- Sie beinhaltet Vorschriften zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Verkehr der Daten.
- Sie schützt Grundrechte und Grundfreiheiten natürlicher Personen und insbesondere deren Recht auf Schutz personenbezogener Daten.
- Sie soll sicherstellen, dass der freie Verkehr personenbezogener Daten in der EU aus Gründen des Schutzes natürlicher Personen bei der Verarbeitung

personenbezogener Daten weder eingeschränkt noch verboten wird.

Siehe hierzu auch den Gesamtumfang des Gesetzes unter [dsgvo-gesetz.de](https://www.dsgvo-gesetz.de), veröffentlicht durch das Unternehmen Intersoft Consulting ([intersoft-consulting.de](https://www.intersoft-consulting.de)).

Der Sicherheitsschild

Jedes Unternehmen sollte sich die entscheidende Frage stellen: Welche Unternehmensdaten sind geschäftskritisch bzw. sind von existenzieller Bedeutung? Im Hinblick auf eine effektive »Sicherheitspolitik« ist es von zentraler Bedeutung, wenn Maßnahmen und Investitionen für IT-Sicherheit auf genau diese kritischen Daten fokussiert werden können. Bei einer solchen gezielten Vorgehensweise können in nicht unerheblichem Maße Kosten gespart werden.

Beispiel: Produkthaftungsdaten und digitalisierte Ausgangsrechnungen sind im Laufe von mehreren Jahren auf verschiedene Server im Unternehmensnetzwerk verteilt abgespeichert worden. Darüber hinaus liegen Kopien dieser Daten in der Cloud bei einem externen Provider, da diese mit einer speziellen Software verarbeitet bzw. archiviert wurden. Es ist nun unmittelbar einsehbar, dass eine Absicherung dieser verteilt abgelegten Daten relativ schwierig bzw. umfangreich

ist. Hier schafft eine zentrale Datenhaltung Abhilfe. Werden sämtliche Daten beispielsweise in der Cloud beim externen Provider gehalten und dieser damit beauftragt (unter Einhaltung gesetzlicher Vorgaben), die erforderlichen Sicherheitsleistungen zu erbringen, so wird neben erhöhter Effektivität (sensible Daten an einem einzigen physischen Ort zu schützen, lässt sich wesentlich besser realisieren als bei einem Verbleib in dezentralen Speichermedien) auch eine sichtbare Kostenoptimierung erreicht.

Abbildung 7-17 zeigt in einem möglichen Beispiel einen diese Aspekte berücksichtigenden »Sicherheitsschild« mit den zu schützenden Unternehmensdaten im Zentrum, die von geeigneten Prozessen und Technologien umgeben einen optimalen Schutz ermöglichen können.

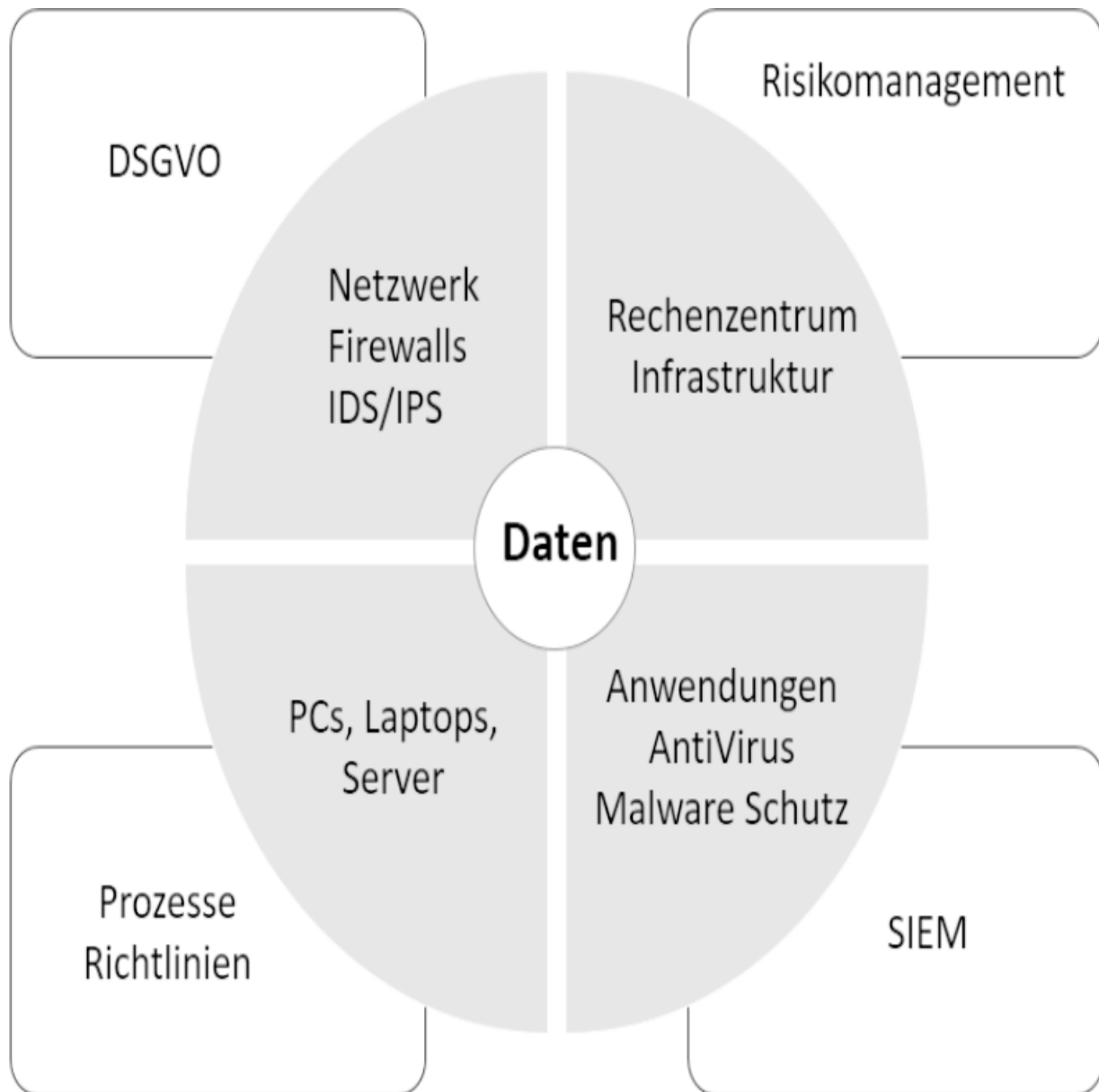


Abb. 7–17 *Beispiel eines möglichen Sicherheitsschildes*

Die hier aufgeführten Disziplinen sind beispielhaft zu verstehen und erheben keinerlei Anspruch auf Vollständigkeit. Sie sollen vielmehr darauf hinweisen, dass ein datenzentrisches Sicherheitsmodell auf einem durchaus komplexen

konzeptionellen Ansatz beruht und dieser verschiedene Disziplinen in einem Unternehmen einbeziehen muss.

Troubleshooting in IP-Netzwerken

Wie in den vorhergehenden Kapiteln mehrfach dargestellt, ist TCP/IP in der EDV-Welt das bekannteste und am weitesten verbreitete Übertragungsprotokoll. Daraus ergibt sich auch, dass es das Protokoll ist, das die meisten Kontrollmöglichkeiten für den Fehlerfall benötigt. Ein Netzwerk, das beispielsweise von vornherein überdimensioniert wurde (drastisch überhöhte Bandbreite), ist in der Lage, zahlreiche Probleme zu »schlucken« und sie nicht sichtbar werden zu lassen. Wenn sich jedoch die Anzahl der Arbeitsstationen verdreifacht oder vervierfacht, ohne dass die bestehende Problematik analysiert und die Störungen behoben wurden, führt dies unweigerlich zu »merkwürdigen Effekten«. Bei einer durchschnittlichen echten Netzwerklast von 20 % könnte die Gesamtlast, inklusive des *Error Traffic*, leicht auf 40 bis 50% steigen. Damit wird ein Status erreicht, der die Netzwerkqualität deutlich beeinträchtigen kann. Ein ähnlicher Effekt ist zu beobachten, wenn ein 100-Mbit/s-Fast-Ethernet-Segment plötzlich mit Anwendungen zur Realisierung von Videokonferenzen belastet wird. Die angenommene, aber durchaus realistische Bandbreite von etwa 128 kbit/s pro Benutzer ergäbe in einem größeren Netzwerk bei 100 parallel aktiven Benutzern eine zusätzliche Beanspruchung von über 10%.

Um eine möglichst optimale Netzwerkleistung zu erzielen, ist es erforderlich, das Netzwerk kontinuierlich zu beobachten, um bei Auftreten bestimmter Symptome rechtzeitig gegensteuern zu können. Dazu sind einerseits allgemeine Kenntnisse über charakteristisches Netzwerkverhalten (aus der Praxis) erforderlich und andererseits die Verfügbarkeit leistungsfähiger Hilfsmittel, die zur Netzwerkanalyse eingesetzt werden können. Beide Bedingungen sind eigentlich untrennbar miteinander verbunden, denn die Sammlung konkreter und vor allem analysierter Erfahrungen lässt sich vorzugsweise mit einschlägigen Analyse-Tools realisieren. Speziell bei der Diagnose und Fehlersuche stellt TCP/IP einige Zusatzoptionen und Dienste zur Verfügung, die das Auffinden und ggf. die Beseitigung entsprechender Fehler oder Probleme wesentlich vereinfachen können. Neben so einfachen Dingen wie PING oder TRACEROUTE gehören dazu auch spezielle Dienste bzw. Anweisungen wie ARP, NETSTAT oder IFCONFIG (IPCONFIG), die wichtige Angaben zu dem Status der TCP/IP-Verbindungen liefern können.

Analysemöglichkeiten

Vor der Fehlersuche muss zunächst einmal eine Analyse des Fehlers bzw. der Fehler erfolgen. Wann tritt das Problem auf? Kann man es auf bestimmte Geräte oder Komponenten

eingrenzen? Tritt das Problem nur zu bestimmten Zeiten auf? Lässt es sich reproduzieren? All das sind Fragen, für deren Antworten eine sorgfältige Analyse erforderlich ist und in einer Netzwerkkumgebung verschiedene Hilfsmittel zur Verfügung stehen.

Der Netzwerk-Trace

Um Netzwerk-Monitoring und Netzwerkanalyse zeitlich einzuordnen, erfolgt die Analyse in der Regel *nach* dem Monitoring, was natürlich zunächst nicht weiter verwunderlich erscheint. Es gibt aber auch analytisch konzipierte Monitore, die bereits während der Beobachtung bestimmte Netzcharakteristika als Symptome identifizieren können. Treten solche Symptome auf, so kann anschließend ein gezielter *Trace* (Ablaufverfolgung) aktiviert werden, der letztlich Detailinformationen über eine bestimmte Kommunikationssituation liefern soll.

Ebenso wie ein Netzwerk-Monitoring den individuellen »Normalzustand« eines Netzwerks erst durch Langzeitmessungen ermitteln muss, besteht bei der Netzwerkanalyse der Bedarf nach einer umfangreichen Sammlung störungsfreier Kommunikationsmuster:

- TCP-Session-Aufbau und -Abbau

- Ablauf einer FTP-Session
- Ablauf einer TELNET-Session
- Session-Aufbau und -Abbau bei 3270-Ressourcen
- Einfügen einer Token-Ring-Station in den Ring
- Routenverfolgung bei SRB
- Erfassung der Charakteristika verschiedener Protokolle
- Kommunikationsverhalten der Routing-Protokolle (OSPF, RIP usw.)
- Charakteristika eines HTTP-Datenstroms
- DNS-Namensauflösung
- usw.

Die verfügbaren Kommunikationsmuster werden bei Traces, die eine Fehler- oder Störsituation aufzeichnen, immer wieder zu Vergleichszwecken herangezogen. Dabei lässt sich ein *Netzwerk-Trace* grundsätzlich als die Aufzeichnung einer Kommunikation verstehen, wie sie zwischen zwei oder mehr Kommunikationspartnern stattgefunden hat. Oft werden *Traces* nur zur Beurteilung bestimmter Passagen einer Kommunikation benötigt, beispielsweise für die Analyse des Beginns einer Konversation (Session-Aufbau). Da es zahlreiche verschiedenartige Netzwerkprotokolle, sprich »Gesprächsregeln«, gibt, sieht die Kommunikation zwischen zwei Partnern über das TCP-Protokoll völlig anders aus als beispielsweise über SNA oder NetBIOS.

Darüber hinaus werden *Traces* für die Überprüfung von Datenströmen eingesetzt. Für die Beurteilung von Problemsituationen sind grundsätzlich nicht nur die Kommunikationsregeln der Netzwerkprotokolle relevant, sondern auch die Daten selbst können einen wertvollen Beitrag zur Netzwerkanalyse leisten. Stellt man durch einen Trace beispielsweise fest, dass sich der Inhalt eines Datensatzes nach Versand durch die Sendestation im Verlauf des Transports durch das Netzwerk verändert hat, so wird der bereits vorgenommene Trace verfeinert. Durch eine Detailbetrachtung jedes einzelnen Frames, jedes Bytes und jedes Bits soll dann ermittelt werden, welches Ereignis bzw. welcher Prozess für den Kommunikations- bzw. Protokollfehler verantwortlich gemacht werden kann.

Bei der Erstellung von *Traces* ist die Einrichtung von Filtern besonders wichtig. Abstrakte, nicht näher spezifizierte Trace-Läufe nehmen eine extrem große Menge an Kommunikationsdaten auf und verwischen damit möglicherweise den Blick auf die entscheidenden Phasen. Restriktive Trace-Optionen führen zu einer wesentlich genaueren Aufzeichnung und damit zu einer schnelleren Analyse des fokussierten Kommunikationsverhaltens.

Bei der Anschaffung von Monitoring- und Analyse-Tools ist zu überlegen, ob die Installation eines einzigen Tools ausreicht oder ob an verschiedenen Stellen im Netzwerk entsprechende Komponenten positioniert werden müssen. Um zwei miteinander kommunizierende Arbeitsstationen zu beobachten, die sich in unterschiedlichen Netzwerksegmenten befinden, müssen schon zwei Analyse-Tools eingerichtet werden: eines im Segment von Arbeitsstation A und eines im Segment von Arbeitsstation B. Erst dann kann der vollständige Datenfluss zwischen den Kommunikationspartnern *lückenlos* über *alle* Kommunikationsschichten hinweg aufgezeichnet werden (siehe [Abb. 8-1](#)).

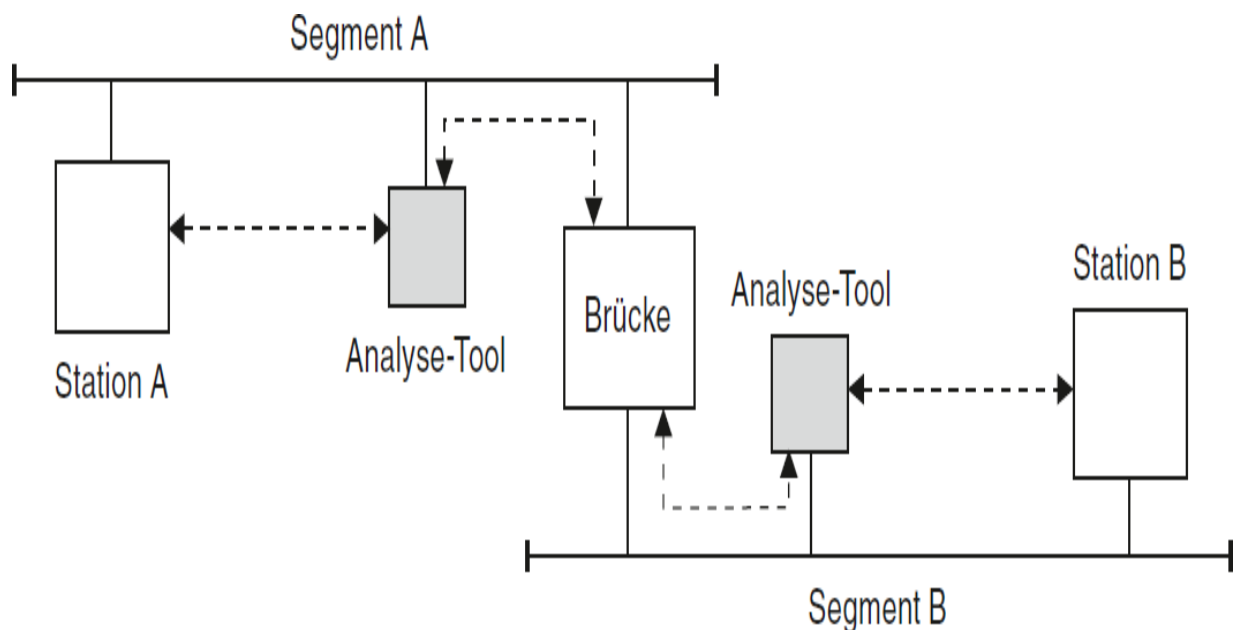


Abb. 8-1 Mehrere Analyse-Tools in einem Netzwerk

Sendet in dem hier dargestellten Beispiel die Station A ein Datenpaket (*Frame*), so erhält dieses das Analyse-Tool aus Segment A und dokumentiert somit den Kommunikationsstatus *vor* Segmentübergang an der Brücke. Die Brücke kopiert den Frame und gibt ihn an Segment B ab. Dort wird der Frame vom Analyse-Tool erfasst, bevor er an die Zielstation B übertragen wird. Somit ist auch bekannt, wie der Frame *nach* Segmentübergang »aussieht«. Analog erfolgt die Erfassung in Gegenrichtung.

Würde hingegen lediglich ein einziges Analyse-Tool installiert (siehe [Abb. 8-2](#)), so könnten die Daten-Frames des benachbarten Segments B bis zur Brücke nicht zweifelsfrei analysiert werden (dies gilt zumindest für Layer-2-Kommunikation). Nur diejenigen Frames werden vom Analyse-Tool »gesehen«, die von der Brücke »durchgelassen« werden. Potenzielle Fehler-Frames bleiben beispielsweise auf der Strecke.

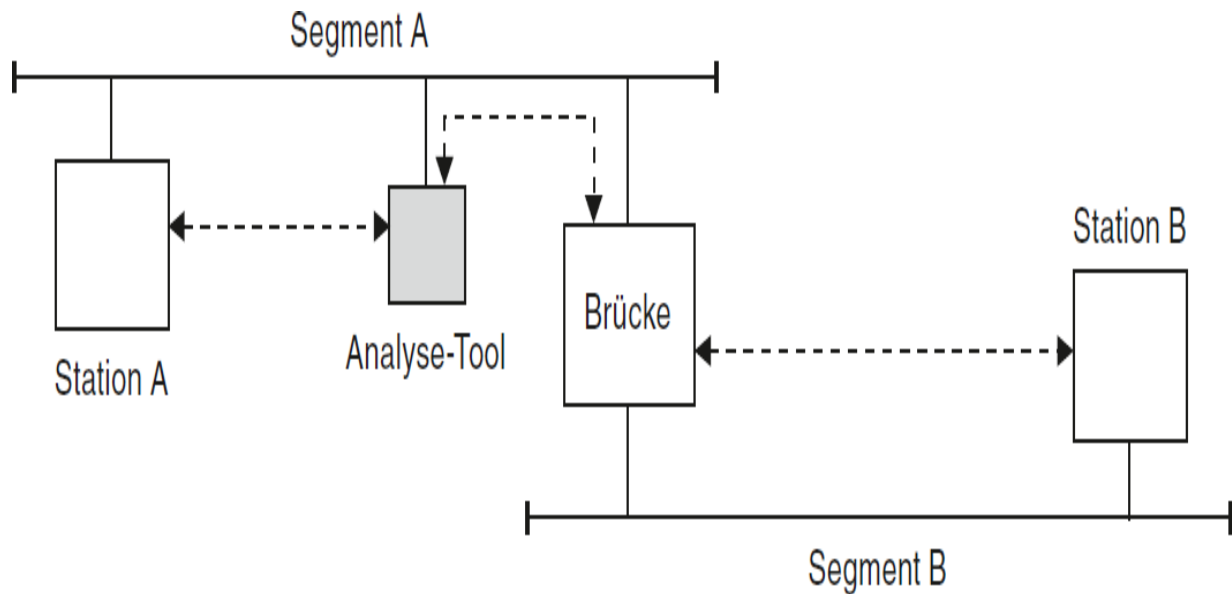


Abb. 8–2 *Einsatz eines einzelnen Analyse-Tools*

HINWEIS

Die Einrichtung fest installierter Analyse-Tools als Standgeräte sollte nur dort vorgenommen werden, wo eine kontinuierliche Beobachtung oder Analyse erforderlich ist. Für wechselnde Einsatzorte ist die Verwendung mobiler Geräte zu empfehlen.

Netzwerkstatistik

Die meisten Analyse-Tools bieten ebenfalls integrierte Möglichkeiten zur Durchführung statistischer Messungen. Dabei wird der Datenstrom »Frame für Frame« nach Protokoll-Headern überprüft und nach der Häufigkeit ihres Auftretens zu Protokollkategorien kumuliert. So ergibt sich nach einer

angemessenen Kumulationszeit ein relativ genaues Bild des Protokollspektrums im Netzwerk bzw. Netzwerksegment.

Das Protokollprofil eines Netzwerks liefert für Design, Management und Analyse wichtige Informationen. Messungen, die über einen längeren Zeitraum durchgeführt werden, ermöglichen beispielsweise Aussagen über bevorstehende Bandbreitenengpässe, sodass rechtzeitig Maßnahmen zur Optimierung ergriffen werden können. Außerdem bieten aktuelle Statistikmessungen einen kontinuierlichen Überblick über die Protokollsituation im Netzwerk. Wird beispielsweise irgendwo im Netzwerk ein Rechner mit einem bislang fremden Protokoll ins Netz »gehängt«, so wird dies sofort bemerkt. Darüber hinaus können damit ebenfalls wichtige Informationen hinsichtlich eines wechselnden Lastverhaltens ermittelt werden:

- Zu welchen Tageszeiten treten Lastspitzen auf?
- Treten die »Spitzen« regelmäßig auf?
- Ist eine Kommunikationsverteilung durch organisatorische Maßnahmen möglich?

Remote Network Monitoring (RMON)

Grundlage des RMON (*Remote Monitoring*) ist eine gegenüber der MIB II (*Management Information Base II*) um zahlreiche

Groups und *Managed Objects* erweiterte RMON-MIB. Auf Basis einer Client-Server-Kommunikation erfolgt ein kontinuierlicher Austausch von Managementinformationen zwischen dem RMON-Agenten (auch RMON-Probe genannt) und der Network Management Station.

HINWEIS

Nähere Angaben zu MIB und den Möglichkeiten des SNMP-Protokolls enthält [Kapitel 6](#).

Konzeption

RMON-Komponenten sind in der Regel Hard- und/oder Softwarekomponenten, die dazu verwendet werden, innerhalb eines Netzwerks analytische, aber auch Verwaltungsaufgaben zu übernehmen. Es handelt sich dabei entweder um reine Softwareimplementierungen, die auf geeigneter, bereits vorhandener Hardware eingesetzt werden können, oder um eine speziell entwickelte Hardware-Software-Kombination als Standalone-Gerät in Netzwerken oder Netzwerksegmenten.

Solche *RMON-Probes* werden oft dafür verwendet, entfernte, nicht direkt erreichbare Ressourcen zu beobachten bzw. zu »managen«. Dies geschieht häufig bei Service-Providern, die auf

diesem Wege das geografisch entfernte Netzwerk ihres Kunden betreuen (siehe [Abb. 8-3](#)).

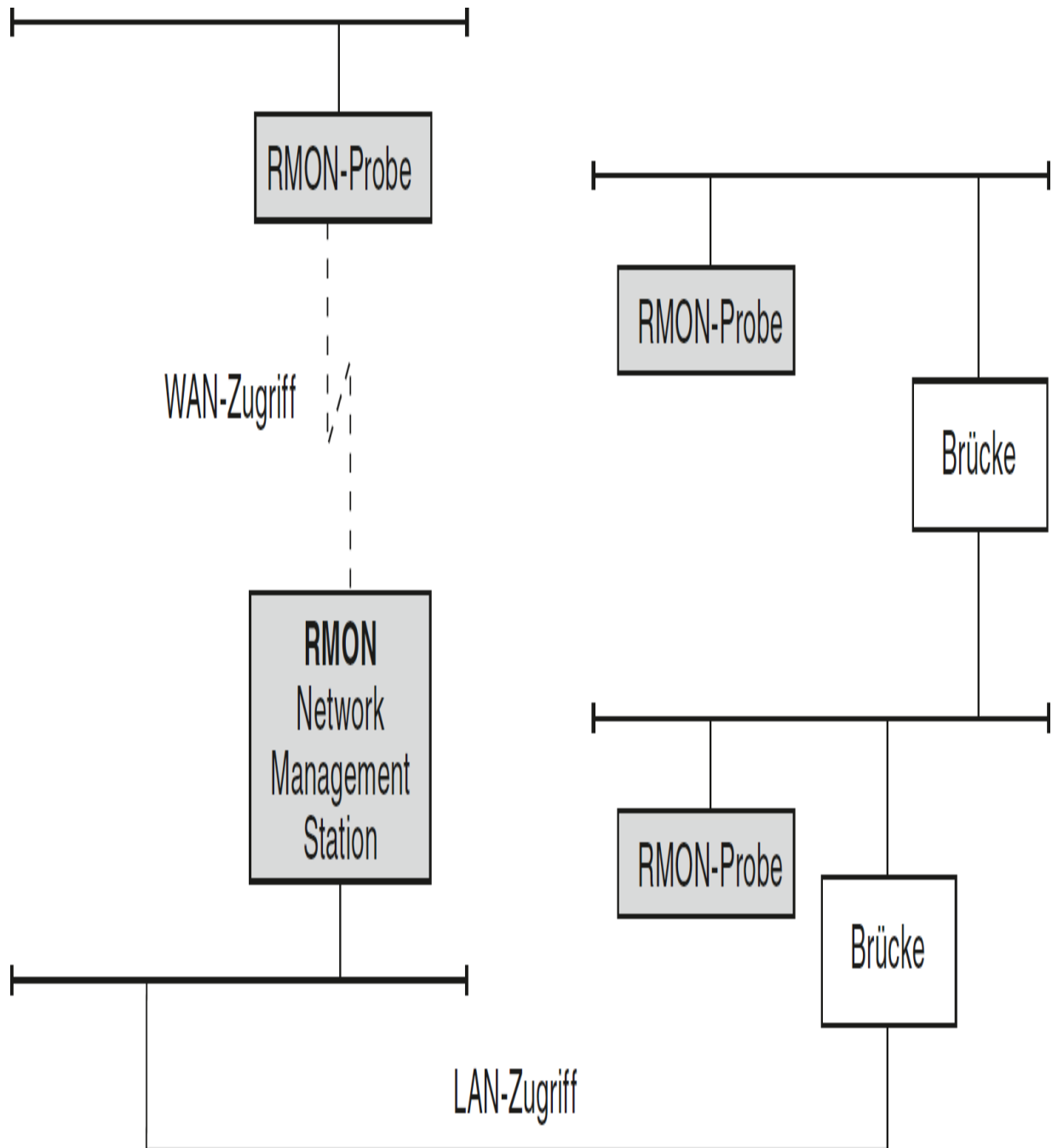


Abb. 8-3 Konzept des Remote-Monitoring

Die RMON-MIB

Als integraler Bestandteil der *Structure and Identification of Management Information* (SMI) geht aus [Abbildung 8-4](#) die Position hervor, an der sich die Gruppen der RMON-MIB befinden. Die Beschreibung der einzelnen *Managed Objects* innerhalb der Gruppen erfolgt in der *Abstract Syntax Notation One* (ASN/1). Die fünf Beschreibungsfelder *object*, *syntax*, *definition*, *access* und *status* beziehen sich jeweils auf ein einzelnes *Managed Object*.

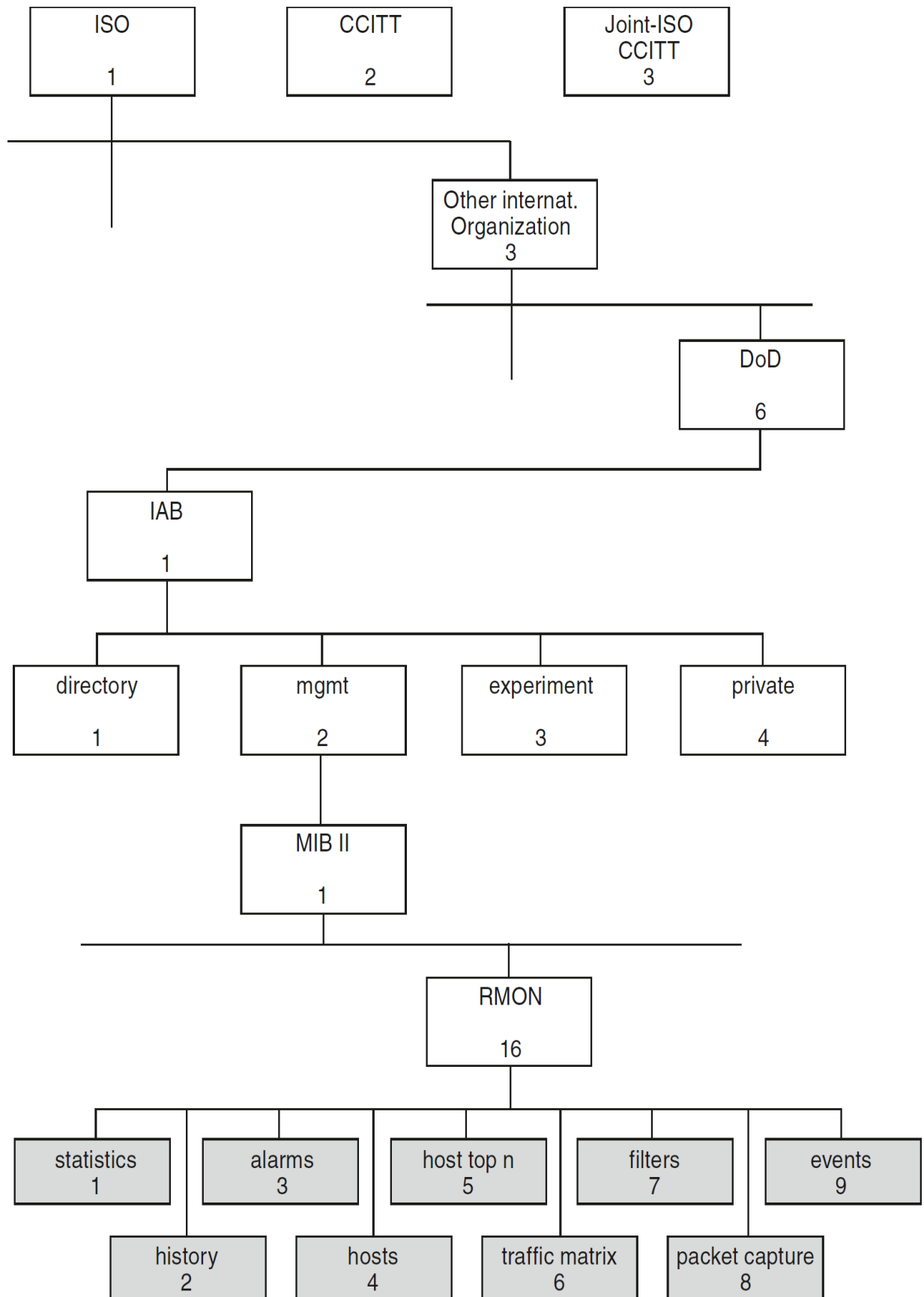


Abb. 8–4 *Gruppen der RMON-MIB*

Die einzelnen Objekte werden in neun Gruppen zusammengefasst:

- *The Ethernet Statistics Group*

Innerhalb dieser Gruppe ermitteln die einzelnen Probes Statistikinformationen eines jeden Ethernet-Netzwerk-Interface.

- *The History Control and Ethernet History Group*

Diese Gruppe steuert periodisches Sampling der Statistics Group. Die einzelnen Sampling-Intervalle sind konfigurierbar.

- *The Alarms Group*

Diese Gruppe realisiert einen Mechanismus, der bei Auslösung bestimmter definierter Schwellenwerte Alarme generiert. Für die Implementierung dieser Gruppe ist außerdem die *Events Group* erforderlich.

- *The Hosts Group*

In dieser Gruppe werden zu jedem Host innerhalb eines Segments Statistikdaten gesammelt, z.B. über versendete und empfangene Pakete bzw. Bytes, Broadcasts, Multicasts usw.

- *The Host Top N Group*

Hier können sortierte Statistiken, die auf der *Hosts Group* basieren, erzeugt werden.

- *The Traffic Matrix Group*

In dieser Gruppe lässt sich eine Verkehrsmatrix für die Host-Host-Kommunikation erstellen. Die einzelnen Ressourcen werden anhand ihrer MAC-Adressen identifiziert.

- *The Filters Group*

Diese Gruppe ermöglicht die Definition von Paketfiltern, die nach unterschiedlichen Kriterien ausgewählt werden können. Durch logische Verknüpfungen (AND, OR, NOT, XOR, EQUAL, NOT-EQUAL) können die Filter auch untereinander kombiniert werden.

- *The Packet Capture Group*

Diese Gruppe lässt eine Definition der Trace-Buffer gemäß den konfigurierten Parametern innerhalb der *Filters Group* zu.

- *The Events Group*

Die innerhalb der übrigen RMON-MIB definierten Schwellenwerte sind für die Generierung von Events verantwortlich. Diese protokollieren die Ereignisse entweder lokal oder sie bilden SNMP-Traps, die über das Netzwerk zur NMS übertragen werden. Die Events werden mit einem Time Stamp und einer erklärenden Beschreibung versehen.

Weiterführende Informationen zu Remote Monitoring MIBs sind den einschlägigen RFCs zu entnehmen.

Analyse in Switched LANs

Während die Analyse in einem sogenannten *Shared Media LAN* relativ unproblematisch ist, indem dort das Analyse-Tool gemeinsam mit den anderen Stationen in das gemeinsam genutzte Netzwerksegment »gehängt« und »mitgehört« wird, welcher Datenverkehr auf dem Medium anliegt, gestaltet sich die Herstellung aussagefähiger Analysen in einem *Switched LAN* weitaus schwieriger. Hier reicht es nicht, das Analyse-Tool mit einem beliebigen Switch-Port zu verbinden und den dort anliegenden Datenverkehr zu beobachten, denn dieser Port stellt ja lediglich diejenigen Daten bereit, die für das angeschlossene Endgerät bestimmt sind (einschließlich aller erforderlichen Broadcasts). Um den gesamten Datenverkehr bzw. einen bestimmten Datenfluss zu beobachten und zu Analyse Zwecken aufzuzeichnen, stehen grundsätzlich folgende Möglichkeiten zur Verfügung:

- Anschluss des Analyse-Tools an einen sogenannten *Mirror-Port*; dieser spezielle Switch-Port kann über die Switch-Software so konfiguriert werden, dass er den gesamten Datenfluss einer anderen auf dem Switch anliegenden Station übergibt.
- Anschluss eines *Media Taps*. Dabei wird der relevante Datenverkehr einfach durch diesen Tap geschleust und über

einen separaten Port, an den das Analyse-Tool angeschlossen wird, mitgelesen. Ein *Media Tap* kann mehrere parallele Full-Duplex-Verbindungen über jeweils separate Durchgangsports abdecken, wobei das Analyse-Tool dann gleichzeitig auf diese Verbindungen zugreifen kann.

Während die eingeschränkte Leistungsfähigkeit älterer Switches ein Mitlesen des Datenverkehrs auf dem Gerät selbst nur unter Performance-Einbußen zulässt, ermöglicht neuere, leistungsfähigere Hardware das »Sniffen« auf dem Switch selbst heute in der Regel ohne Probleme. Es sollte aber auch hier darauf geachtet werden, dass vorher möglichst genau festgelegt wird, welcher Datenverkehr analysiert werden soll, denn eine zusätzliche Belastung der CPU bedeutet diese Form der Analyse allemal. Oft ist es auch sinnvoll, den mit Datenpaketen gefüllten Cache-Speicher des Switches in einer Datei abzuspeichern, die dann später zu Analysezwecken per FTP/TFTP vom Switch auf einen Analyserechner übertragen wird.

Verbindungstest mit PING

Nach den eher theoretischen Überlegungen der vorhergehenden Abschnitte sollen nachfolgend ganz konkrete Möglichkeiten zur Fehler- und Problemanalyse vorgestellt

werden. So ist die generelle Grundvoraussetzung für die Übertragung von Daten in einem IP-Netzwerk, dass eine Hardwareverbindung zu einem anderen System (per Netzwerkkomponenten und Kabel) besteht und die notwendige Konfiguration (Endgeräte müssen über einen TCP/IP-Protokollstack verfügen) durchgeführt worden ist.

Sobald dies gewährleistet ist, besteht in einem solchen Netzwerk mittels der Anweisung PING eine sehr einfache und schnelle Möglichkeit, die generelle Verbindung zwischen zwei Endgeräten zu überprüfen. PING ist eine Anwendung bzw. Anweisung, die auf den unteren Schichten des OSI-Referenzmodells angesiedelt ist und eine Überprüfung der physikalischen Verbindung sowie der IP-Konfiguration ermöglicht.

HINWEIS

Mit PING kann nicht nur die Verbindung zwischen zwei IP-Endgeräten überprüft werden, sondern auch die IP-Konfiguration des lokalen Endgeräts erfolgen.

Selbsttest

PING ist ein Dienst bzw. Hilfsprogramm, mit dessen Hilfe die Erreichbarkeit auf IP-Ebene überprüft werden kann. Dabei

wird zunächst ein ARP-Broadcast (ARP = *Address Resolution Protocol*) versendet, um die MAC-Adresse des Endgeräts herauszufinden. Nachdem die Antwort des Zielrechners eingetroffen ist, wird an die MAC-Adresse des anderen Endgeräts eine Echoanforderung per ICMP (*Internet Control Message Protocol*) geschickt. Dieses beantwortet die Anforderung (Request) mit *ICMP-Echo-Reply-Paketen*.

Mit PING kann jedoch nicht nur die Verbindung zu anderen Endgeräten getestet, sondern auch jederzeit die Konfiguration des lokalen Rechners überprüft werden. Auf diese Art besteht die Möglichkeit, etwaige Fehlerquellen des lokalen Rechners auszugrenzen.

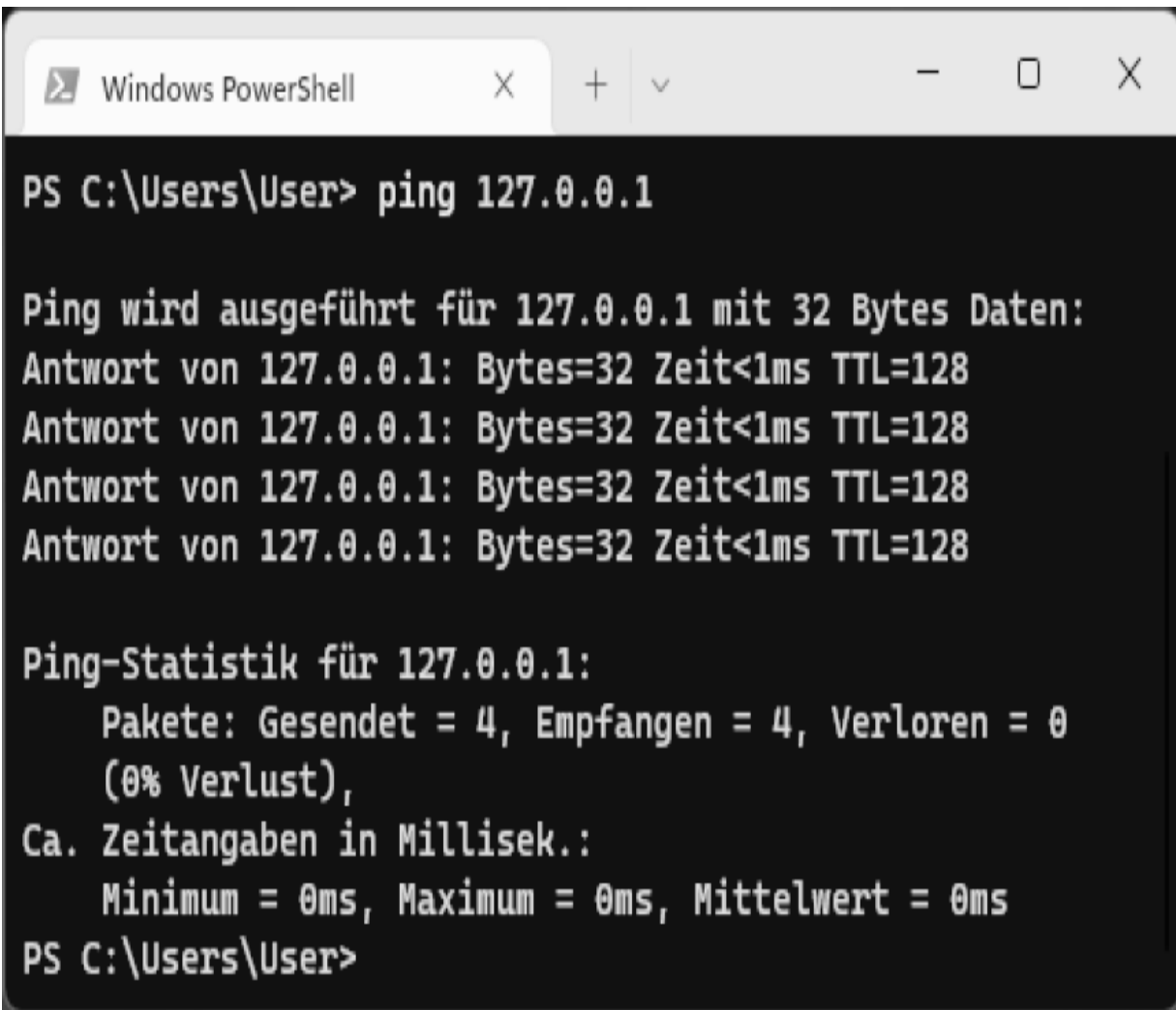
Sobald Probleme mit der Verbindungsaufnahme zu einem anderen Rechner auftreten, sollten zunächst einmal der lokale Rechner und dessen Netzwerkkonfiguration (IP-Konfiguration) überprüft werden. Zu diesem Zweck kommt die *Loopback-Adresse* bzw. *Loopback-Schnittstelle* zum Einsatz. So kann mit einer der beiden folgenden Anweisungen die Konfiguration des aktuellen Rechners überprüft werden:

```
ping loopback
```

oder alternativ

```
ping 127.0.0.1
```

Während bei der ersten Anweisung der Name *loopback* eingesetzt wird (dieser wird in der Hosts-Datei aufgelöst), wird bei der zweiten Anweisung die tatsächliche Loopback-Adresse eingesetzt (127.0.0.1). [Abbildung 8–5](#) zeigt als Ergebnis einer erfolgreich durchgeführten IP-Konfiguration auf dem lokalen Windows-Rechner nach Eingabe des zweiten Kommandos (Windows-Beispiele erfolgen stets unter Windows 10/11, Beispiele mit macOS unter Version 12.6 und für die Darstellung unter UNIX wird Debian 10 mit einem Raspberry Pi 4 verwendet):

A screenshot of a Windows PowerShell window. The title bar shows 'Windows PowerShell' with standard window controls. The command prompt shows 'PS C:\Users\User> ping 127.0.0.1'. The output displays four successful ping responses, each with 'Bytes=32', 'Zeit<1ms', and 'TTL=128'. It also includes a 'Ping-Statistik für 127.0.0.1:' section showing 'Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust)' and 'Ca. Zeitangaben in Millisek.: Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms'. The prompt ends with 'PS C:\Users\User>'.

```
PS C:\Users\User> ping 127.0.0.1

Ping wird ausgeführt für 127.0.0.1 mit 32 Bytes Daten:
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128
Antwort von 127.0.0.1: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 127.0.0.1:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
PS C:\Users\User>
```

Abb. 8–5 *Ergebnis aus der Eingabe: ping 127.0.0.1*

HINWEIS

Wenn ein PING auf die Loopback-Adresse des lokalen Rechners (Systems) erfolgreich durchgeführt wurde, kann davon ausgegangen werden, dass das lokale System für den Einsatz von TCP/IP korrekt konfiguriert worden ist.

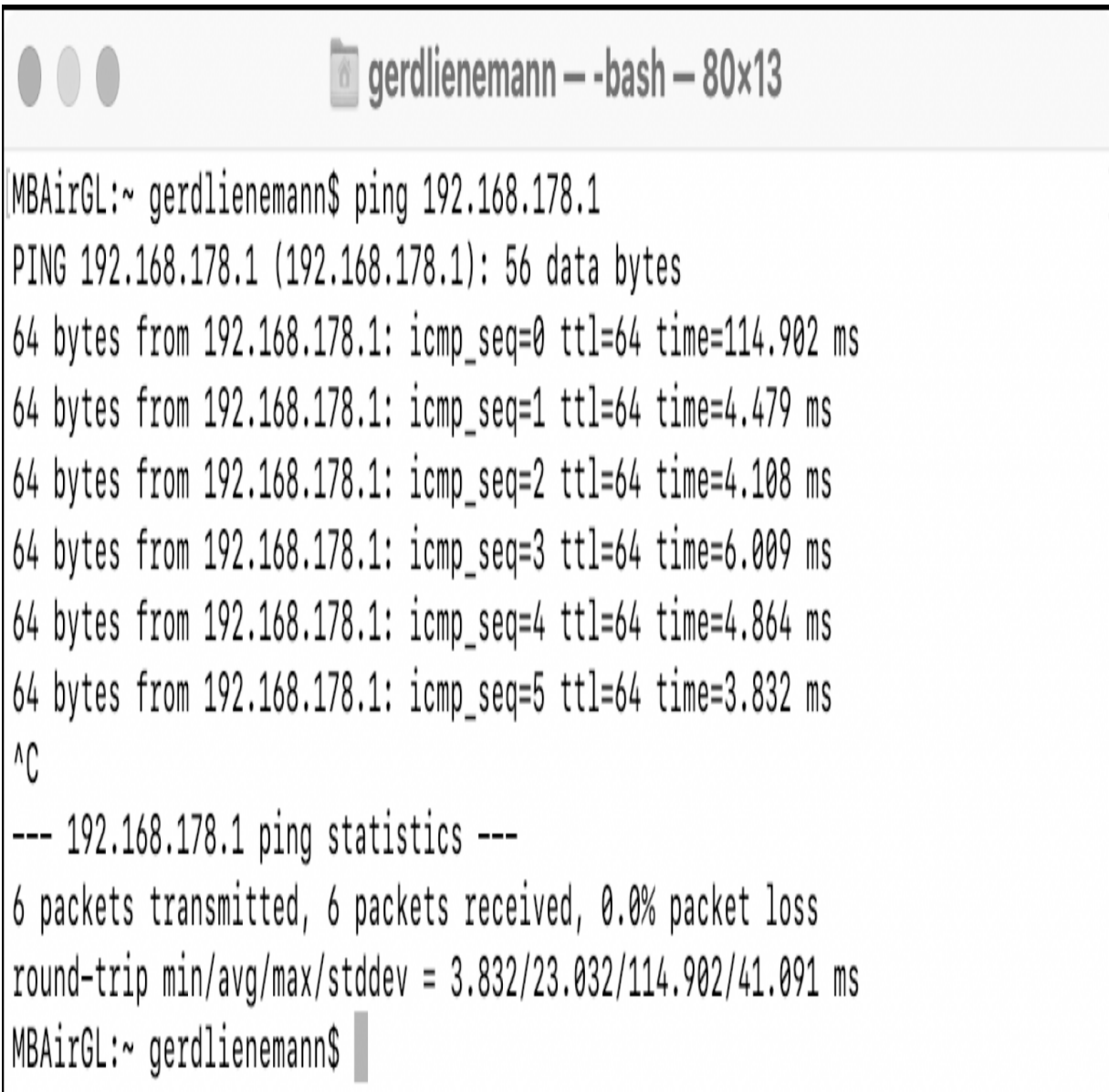
Erfolgt ein »ping« auf den Namen »loopback«, so muss sichergestellt sein, dass der Name in der *hosts*-Datei auch konfiguriert ist.

Test anderer Endgeräte

Die einfachste Form der Überprüfung einer Verbindung zwischen unterschiedlichen IP-Endgeräten erfolgt mit der PING-Anweisung auf die folgende Art:

```
ping 192.168.178.1
```

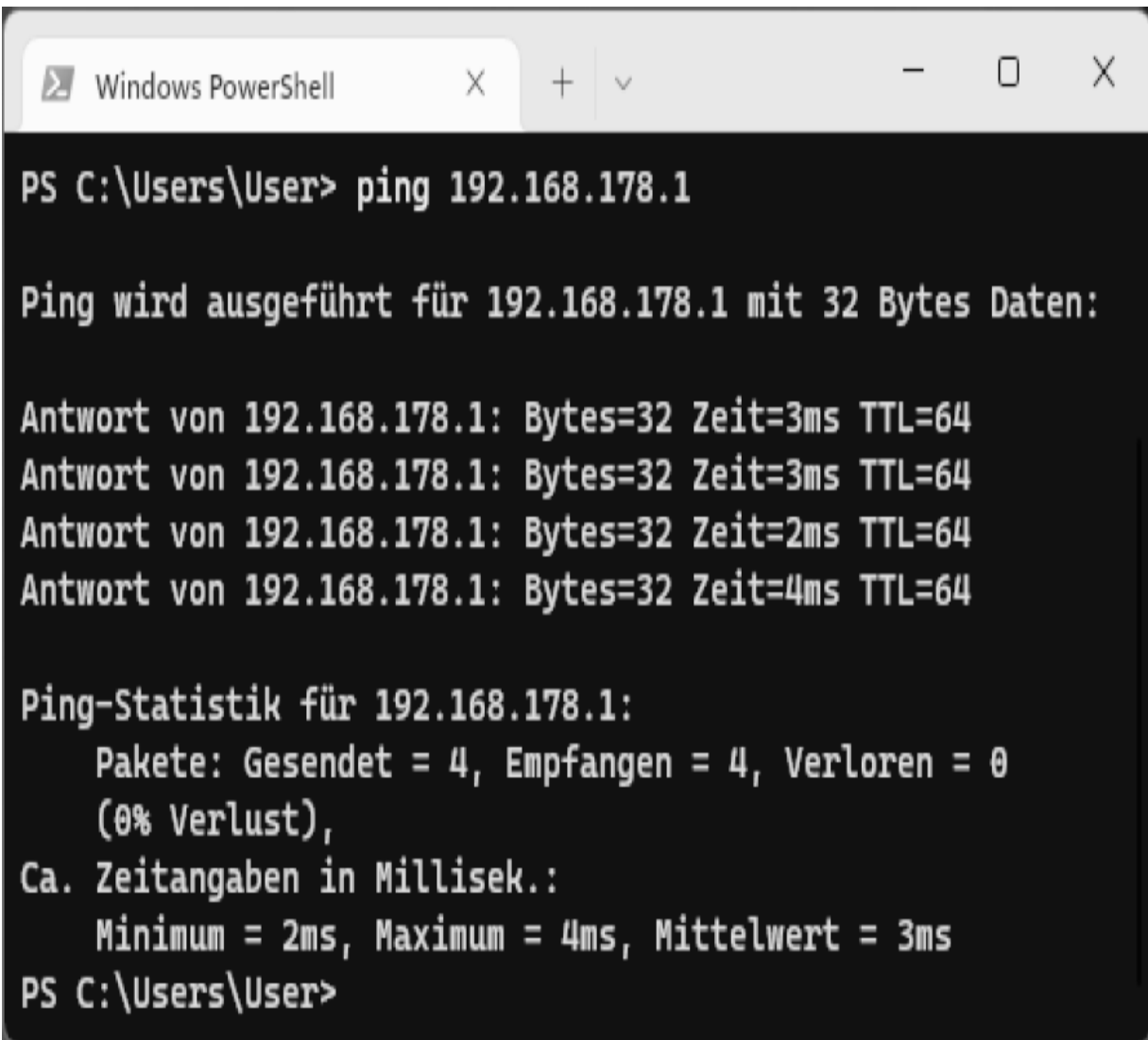
Damit wird von dem lokalen Rechner eine Verbindung zu dem Rechner mit der IP-Adresse 192.168.178.1 aufgebaut. Es werden dabei Datenpakete gesendet, die von dem anderen System bestätigt werden müssen. Sobald die Verbindung hergestellt werden kann, erscheinen am Bildschirm Angaben der Form wie in [Abbildung 8-6](#), sofern es sich um ein macOS-System handelt.

A screenshot of a macOS terminal window. The title bar at the top shows three window control buttons (red, yellow, green) on the left and a folder icon followed by the text "gerdlienemann — -bash — 80x13" on the right. The terminal content shows a user at the "MBAirGL:~ gerdlienemann" prompt typing "ping 192.168.178.1". The output shows a successful ping with 56 data bytes. Six individual ping results are listed, showing times ranging from 3.832 ms to 114.902 ms. The user then presses Ctrl-C (^C), and the terminal displays "ping statistics" showing 0% packet loss and round-trip statistics. The prompt returns to "MBAirGL:~ gerdlienemann\$".

```
MBAirGL:~ gerdlienemann$ ping 192.168.178.1
PING 192.168.178.1 (192.168.178.1): 56 data bytes
64 bytes from 192.168.178.1: icmp_seq=0 ttl=64 time=114.902 ms
64 bytes from 192.168.178.1: icmp_seq=1 ttl=64 time=4.479 ms
64 bytes from 192.168.178.1: icmp_seq=2 ttl=64 time=4.108 ms
64 bytes from 192.168.178.1: icmp_seq=3 ttl=64 time=6.009 ms
64 bytes from 192.168.178.1: icmp_seq=4 ttl=64 time=4.864 ms
64 bytes from 192.168.178.1: icmp_seq=5 ttl=64 time=3.832 ms
^C
--- 192.168.178.1 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 3.832/23.032/114.902/41.091 ms
MBAirGL:~ gerdlienemann$
```

Abb. 8–6 Ergebnis aus der Eingabe (macOS): ping 192.168.178.1

Auf einem aktuellen Windows-System stellt sich die Anzeige eines erfolgreich durchgeführten Verbindungstests dar wie in [Abbildung 8–7](#) gezeigt.



```
Windows PowerShell
PS C:\Users\User> ping 192.168.178.1

Ping wird ausgeführt für 192.168.178.1 mit 32 Bytes Daten:

Antwort von 192.168.178.1: Bytes=32 Zeit=3ms TTL=64
Antwort von 192.168.178.1: Bytes=32 Zeit=3ms TTL=64
Antwort von 192.168.178.1: Bytes=32 Zeit=2ms TTL=64
Antwort von 192.168.178.1: Bytes=32 Zeit=4ms TTL=64

Ping-Statistik für 192.168.178.1:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 2ms, Maximum = 4ms, Mittelwert = 3ms
PS C:\Users\User>
```

Abb. 8–7 Ergebnis aus der Eingabe (Windows): *ping* 192.168.178.1

HINWEIS

Mit der Angabe *TTL* (*Time To Live*) wird die Lebensdauer eines IP-Pakets bezeichnet. Kann das Paket das Ziel in der vorgegebenen Zeit nicht erreichen, wird es verworfen.

Aus den oben dargestellten Angaben geht hervor, dass die Verbindung besteht und der Versand der Pakete von der Gegenseite bestätigt wurde.

Neben der Angabe einer IP-Adresse kann natürlich auch ein Name eingegeben werden (z.B. ping macGL), sofern eine Namensauflösung per DNS oder Hosts-Datei konfiguriert worden ist. Nähere Angaben dazu enthält [Kapitel 5](#).

Sollte die Gegenstelle nicht erreichbar sein, erscheint eine Ausgabe wie in [Abbildung 8–8](#) bei einem macOS-System gezeigt.

A screenshot of a macOS terminal window. The title bar at the top shows three window control buttons (red, yellow, green) on the left and a folder icon followed by the text "gerdlienemann — -bash — 80x14" on the right. The terminal content shows a user at the "MBAirGL:~ gerdlienemann" prompt typing "ping 192.168.178.33". The output shows "PING 192.168.178.33 (192.168.178.33): 56 data bytes" followed by nine lines of "Request timeout for icmp_seq 0" through "icmp_seq 8". The user then presses Ctrl-C, indicated by "^C". The terminal then displays "--- 192.168.178.33 ping statistics ---" and "9 packets transmitted, 0 packets received, 100.0% packet loss". The prompt "MBAirGL:~ gerdlienemann\$" is shown again with a cursor.

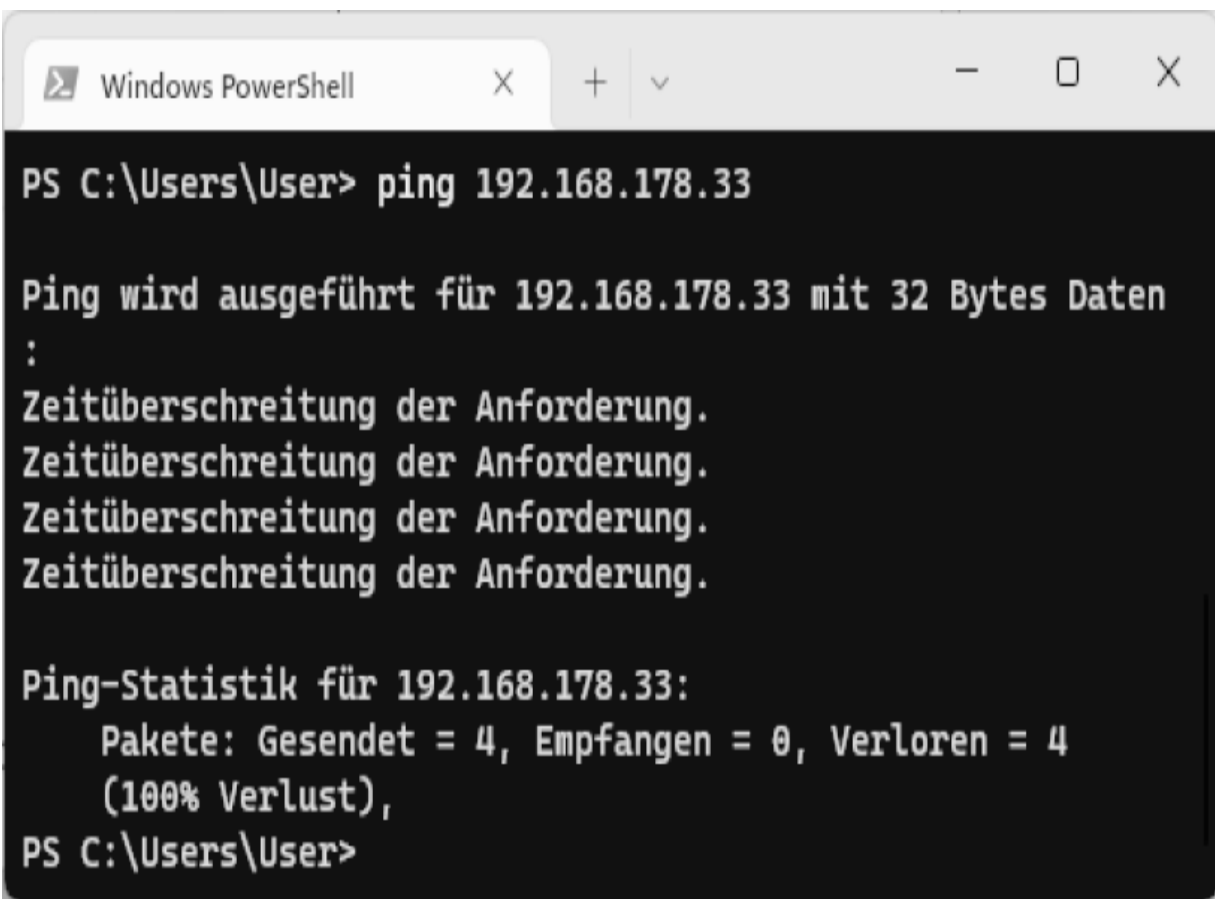
```
MBAirGL:~ gerdlienemann$ ping 192.168.178.33
PING 192.168.178.33 (192.168.178.33): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
Request timeout for icmp_seq 6
Request timeout for icmp_seq 7
^C
--- 192.168.178.33 ping statistics ---
9 packets transmitted, 0 packets received, 100.0% packet loss
MBAirGL:~ gerdlienemann$
```

Abb. 8–8 *Ping-Kommando (macOS): IP-Adresse nicht erreichbar*

Daraus ist ersichtlich, dass Testpakete übertragen wurden, diese jedoch verloren gegangen sind (*100 % packet loss*). Ursache kann entweder eine fehlerhafte IP-Adresse, ein fehlerhafter Host-Name oder auch die Tatsache sein, dass das

angesprochene Endgerät deaktiviert (ausgeschaltet) wurde, die Netzwerkkarte möglicherweise defekt ist oder einfach keine Netzwerkanbindung besteht.

Abbildung 8-9 zeigt dasselbe Ergebnis wie Abbildung 8-8, allerdings unter Windows.

A screenshot of a Windows PowerShell window. The title bar shows 'Windows PowerShell' with standard window controls. The command prompt shows a user at 'C:\Users\User' running 'ping 192.168.178.33'. The output indicates a failure: 'Ping wird ausgeführt für 192.168.178.33 mit 32 Bytes Daten : Zeitüberschreitung der Anforderung.' repeated four times. The statistics show 'Pakete: Gesendet = 4, Empfangen = 0, Verloren = 4 (100% Verlust)'. The prompt returns to 'PS C:\Users\User>'.

```
PS C:\Users\User> ping 192.168.178.33

Ping wird ausgeführt für 192.168.178.33 mit 32 Bytes Daten
:
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.
Zeitüberschreitung der Anforderung.

Ping-Statistik für 192.168.178.33:
    Pakete: Gesendet = 4, Empfangen = 0, Verloren = 4
    (100% Verlust),
PS C:\Users\User>
```

Abb. 8-9 *Ping-Kommando (Windows): IP-Adresse nicht erreichbar*

Praktische Vorgehensweise im Fehlerfall

Werden generell Verbindungsprobleme mit IP-Endgeräten festgestellt, sollte in einem solchen Fall eine bestimmte Vorgehensweise eingeschlagen werden, um dem Fehler auf den Grund zu gehen.

- Zunächst einmal sollte das PING-Signal an das lokale System (Rechner) gesendet werden, um so dessen IP-Konfiguration zu überprüfen (*PING Loopback*).
- Im nächsten Schritt sollte dann eine PING-Anweisung auf die zugewiesene IP-Adresse des Ursprungssystems erfolgen. Dies geschieht mittels einer PING-Anweisung, bei der die lokale IP-Adresse angegeben wird.
- Anschließend sollte dann das andere Endgerät, zu dem es Verbindungsprobleme gibt, mittels PING angesprochen werden.

Somit ergibt sich im Falle eines Fehlers die generelle Vorgehensweise wie folgt:

- PING auf das Loopback-Interface
- PING auf die lokale Netzwerkkarte
- PING auf das andere System (Endgerät)

Wenn in diesem Zusammenhang das PING-Signal mittels IP-Adresse des anderen Endgeräts versendet werden kann, dies jedoch mit dem Host-Namen fehlschlägt, so deutet das darauf hin, dass ein Problem mit der Namensauflösung (DNS, Hosts-Datei usw.) besteht. Dann ist es also kein Problem der Verbindung, sondern ein Konfigurationsproblem der Namensauflösung.

HINWEIS

Mit der Anweisung *ping -?* können jederzeit die verfügbaren Optionen bzw. Parameter angezeigt werden.

Informationen per NETSTAT

Mit der Anweisung NETSTAT stellt der TCP/IP-Protokollstack eine Möglichkeit zur Überwachung der aktuell laufenden Netzwerkfunktionen zur Verfügung. So liefert NETSTAT unter anderem Informationen über Netzwerkschnittstellen und zeigt Routing-Informationen sowie den Status der aktiven Verbindungen an.

Abbildung 8-10 zeigt das Ergebnis der Anweisung NETSTAT mit Option a. Da die zuvor gezeigten Ergebnisse auf Windows und macOS kaum Unterschiede gezeigt haben, wird im

Folgenden auf eine zusätzliche Darstellung für das System macOS verzichtet.

PS C:\Users\User> netstat -a

Aktive Verbindungen

Proto	Lokale Adresse	Remoteadresse	Status
TCP	0.0.0.0:135	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:445	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:5040	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:5357	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49664	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49665	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49666	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49667	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49668	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49669	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49843	DESKTOP-T6C9156:0	ABHÖREN
TCP	0.0.0.0:49844	DESKTOP-T6C9156:0	ABHÖREN
TCP	192.168.178.43:139	DESKTOP-T6C9156:0	ABHÖREN
TCP	192.168.178.43:51595	20.199.120.85:https	HERGESTELLT
TCP	192.168.178.43:51596	13.107.43.12:https	WARTEND
TCP	192.168.178.43:51597	1drv:https	WARTEND

PS C:\Users\User>

Abb. 8–10 *Kommando »netstat -a«*

Hier wurden alle aktiven TCP- und UDP-Ports angezeigt, die der Rechner abhört und damit sofort reagieren kann, wenn ein Verbindungsversuch stattfindet.

Mit der folgenden Anweisung kann beispielsweise die Routing-Tabelle des aktuellen Rechners angezeigt werden:

```
netstat -r
```

Abbildung 8–11 zeigt das Ergebnis (Auszug).

PS C:\Users\User> netstat -r

=====

Schnittstellenliste

10...e2 e1 a9 31 b7 eaMicrosoft Wi-Fi Direct Virtual Adapter
17...e0 e1 a9 31 b7 eaMicrosoft Wi-Fi Direct Virtual Adapter #2
14...e0 e1 a9 31 b7 eaRealtek 8821CU Wireless LAN 802.11ac USB NIC
13...e0 e1 a9 31 b7 ebBluetooth Device (Personal Area Network)
1.....Software Loopback Interface 1

=====

IPv4-Routentabelle

=====

Aktive Routen:

Netzwerkziel	Netzwerkmaske	Gateway	Schnittstelle	Metrik
0.0.0.0	0.0.0.0	192.168.178.1	192.168.178.43	55
127.0.0.0	255.0.0.0	Auf Verbindung	127.0.0.1	331
127.0.0.1	255.255.255.255	Auf Verbindung	127.0.0.1	331
127.255.255.255	255.255.255.255	Auf Verbindung	127.0.0.1	331
192.168.178.0	255.255.255.0	Auf Verbindung	192.168.178.43	311
192.168.178.43	255.255.255.255	Auf Verbindung	192.168.178.43	311
192.168.178.255	255.255.255.255	Auf Verbindung	192.168.178.43	311
224.0.0.0	240.0.0.0	Auf Verbindung	127.0.0.1	331
224.0.0.0	240.0.0.0	Auf Verbindung	192.168.178.43	311
255.255.255.255	255.255.255.255	Auf Verbindung	127.0.0.1	331
255.255.255.255	255.255.255.255	Auf Verbindung	192.168.178.43	311

=====

Ständige Routen:

Keine

IPv6-Routentabelle

=====

Aktive Routen:

If	Metrik	Netzwerkziel	Gateway
----	--------	--------------	---------

Abb. 8–11 *Kommando »netstat -r«*

Mit der Option »-s« der NETSTAT-Anweisung können die Statistiken der diversen Protokolle angezeigt werden (siehe [Abb. 8–12](#)).

```
PS C:\Users\User> netstat -s
```

IPv4-Statistik

Empfangene Pakete	= 3848
Empfangene Vorspannfehler	= 0
Empfangene Adressfehler	= 1
Weitergeleitete Datagramme	= 0
Empfangene unbekannte Protokolle	= 0
Empfangene verworfene Pakete	= 16
Empfangene übermittelte Pakete	= 3995
Ausgabeanforderungen	= 3719
Verworfenen Routingpakete	= 0
Verworfenen Ausgabepakete	= 0
Ausgabepakete ohne Routing	= 0
Reassemblierung erforderlich	= 0
Reassemblierung erfolgreich	= 0
Reassemblierung erfolglos	= 0
Erfolgreiche Datagrammfragmentierung	= 0
Erfolglose Datagrammfragmentierung	= 0
Erzeugte Fragmente	= 0

IPv6-Statistik

Empfangene Pakete	= 3262
Empfangene Vorspannfehler	= 0
Empfangene Adressfehler	= 13
Weitergeleitete Datagramme	= 0
Empfangene unbekannte Protokolle	= 0
Empfangene verworfene Pakete	= 1
Empfangene übermittelte Pakete	= 3353
Ausgabeanforderungen	= 2933
Verworfenen Routingpakete	= 0
Verworfenen Ausgabepakete	= 3

Abb. 8–12 *Kommando »netstat -s«*

HINWEIS

Die Anweisung NETSTAT verfügt über eine Vielzahl von Parametern, die mit der Option »-?« angezeigt werden können.

ROUTE zur Wegekongfiguration

Die Anweisung ROUTE wird in einer IP-Umgebung primär dazu verwendet, die Routing-Tabelle zu modifizieren. Nach Eingabe von ROUTE ? unter Windows erhält man eine Liste aller nutzbaren ROUTE-Befehle und -Parameter, wie [Abbildung 8–13](#) zu entnehmen ist.

```
Windows PowerShell X + v - □ X

PS C:\Users\User> route ?

Ändert die Netzwerkroutingtabellen.

ROUTE [-f] [-p] [-4|-6] Befehl [Ziel]
      [MASK Netzmaske] [Gateway] [METRIC Metrik] [IF Schnittstelle]

-f      Löscht alle Gatewayeinträge in Routingtabellen. Wird dieser
        Parameter zusammen mit einem der Befehle verwendet, werden
        die Tabellen vor der Befehlsausführung gelöscht.

-p      Wird der Parameter mit dem "ADD"-Befehl verwendet, wird eine
        Route unabhängig von Neustarts des Systems beibehalten.
        Standardmäßig werden Routen nach dem Neustart des Systems
        nicht beibehalten. Dieser Parameter wird für alle anderen
        Befehle ignoriert, da diese immer die entsprechenden beständigen
        Routen betreffen.

-4      Die Verwendung von IPv4 wird erzwungen.

-6      Die Verwendung von IPv6 wird erzwungen.

Befehl  Eine der folgenden Optionen:
        PRINT      Druckt eine Route.
        ADD        Fügt eine Route hinzu.
        DELETE     Löscht eine Route.
        CHANGE     Ändert eine vorhandene Route.

Ziel     Gibt den Host an.

MASK     Gibt an, dass der folgende Parameter ein Netzwerkwert ist.

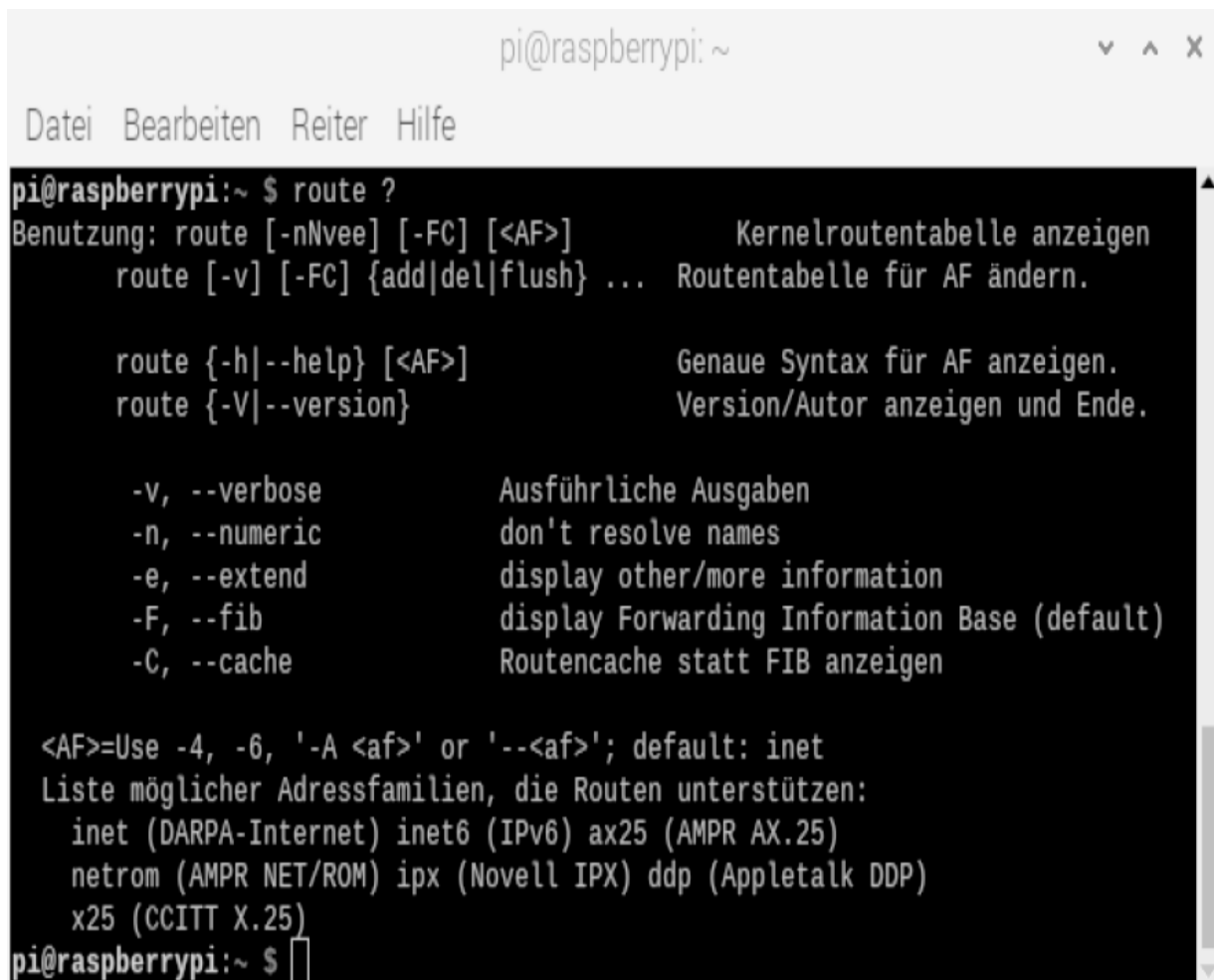
Netzmaske Gibt einen Wert für eine Subnetzmaske für den Routeneintrag an.
        Ohne Angabe wird die Standardeinstellung 255.255.255.255
        verwendet.

Gateway  Gibt ein Gateway an.
```

Abb. 8–13 *Kommando »route ?«*

Dabei liefert diese Anzeige ausführliche Anweisungen, wie das Kommando zu benutzen ist.

Dies ist beispielsweise unter Unix anders. [Abbildung 8–14](#) zeigt die Syntax des ROUTE-Kommandos unter Debian Version 10.



```
pi@raspberrypi:~ $ route ?
Benutzung: route [-nNvee] [-FC] [<AF>]          Kernelroutentabelle anzeigen
             route [-v] [-FC] {add|del|flush} ... Routentabelle für AF ändern.

             route {-h|--help} [<AF>]           Genaue Syntax für AF anzeigen.
             route {-V|--version}              Version/Autor anzeigen und Ende.

             -v, --verbose                     Ausführliche Ausgaben
             -n, --numeric                     don't resolve names
             -e, --extend                     display other/more information
             -F, --fib                         display Forwarding Information Base (default)
             -C, --cache                       Routencache statt FIB anzeigen

<AF>=Use -4, -6, '-A <af>' or '--<af>'; default: inet
Liste möglicher Adressfamilien, die Routen unterstützen:
inet (DARPA-Internet) inet6 (IPv6) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
x25 (CCITT X.25)
pi@raspberrypi:~ $
```

Abb. 8–14 *Befehlssyntax des Route-Kommandos unter Debian*

Der Print-Befehl existiert hier nicht. Unter Debian führt die Eingabe »route -n« zur Anzeige der Routen (siehe [Abb. 8–15](#)).



```
pi@raspberrypi:~  
Datei Bearbeiten Reiter Hilfe  
pi@raspberrypi:~ $ route -n  
Kernel-IP-Routentabelle  
Ziel          Router        Genmask       Flags Metric Ref    Use Iface  
0.0.0.0       192.168.178.1 0.0.0.0       UG    303    0      0 wlan0  
192.168.178.0 0.0.0.0       255.255.255.0 U    303    0      0 wlan0  
pi@raspberrypi:~ $
```

Abb. 8–15 Kommando »route -n« unter Debian 10

Während mit ROUTE PRINT unter Windows die aktuelle Routen-Zuweisung abgerufen werden kann, können mit ROUTE ADD der lokalen Routing-Tabelle neue Routen hinzugefügt und mit ROUTE DELETE bestehende Routen aus der Tabelle entfernt werden.

HINWEIS

Damit zwei Host-Systeme IP-Datagramme austauschen, müssen beide eine Route zum jeweils anderen haben oder ein Standard-Gateway verwenden, dem eine Route bekannt ist. Normalerweise tauschen Router untereinander Informationen mit einem Routing-Protokoll wie RIP (*Routing Information*

Protocol) oder OSPF (*Open Shortest Path First*) aus. Nähere Angaben zum Thema *Routing* enthält [Kapitel 4](#).

Wegeermittlung per TRACEROUTE

Bei TRACEROUTE (TRACERT auf einem Windows-System) handelt es sich um ein Dienstprogramm bzw. eine Anweisung für die Verfolgung aktueller Routen. Es können damit die Knotenpunkte (Router-Hops) zu einem entfernten System angezeigt werden. Die Anweisung verwendet das TTL-Feld (TTL = *Time To Live*) von ICMP-Meldungen (PING), um die Route von einem Host zu einem anderen durch ein Netzwerk zu ermitteln.

Ein Beispiel für die Ausgabe des Befehls TRACEROUTE ist in [Abbildung 8-16](#) dargestellt. Aus einer solchen Anzeige bzw. Ausgabe des TRACEROUTE-Befehls ist ersichtlich, welchen Weg die Datenpakete zu dem angegebenen Ziel-Host nehmen und welche Zeit dafür benötigt wird.

```
gerdlienemann — -bash — 80x24
MBAirGL:~ gerdlienemann$ traceroute www.google.de
traceroute to www.google.de (172.217.16.131), 64 hops max, 52 byte packets
 1 fritz.box (192.168.178.1)  5.046 ms  8.307 ms  3.282 ms
 2 100.68.128.1 (100.68.128.1)  10.941 ms  10.659 ms  7.686 ms
 3 100.127.1.131 (100.127.1.131)  12.641 ms  13.581 ms  15.642 ms
 4 185.22.46.129 (185.22.46.129)  11.443 ms  12.817 ms  9.570 ms
 5 185.22.44.29 (185.22.44.29)  20.543 ms  21.425 ms  21.626 ms
 6 108.170.241.204 (108.170.241.204)  14.030 ms
   108.170.241.205 (108.170.241.205)  22.472 ms
   108.170.241.204 (108.170.241.204)  17.433 ms
 7 216.239.43.36 (216.239.43.36)  21.929 ms
   142.251.236.90 (142.251.236.90)  14.842 ms
   209.85.254.157 (209.85.254.157)  16.556 ms
 8 * 142.251.236.105 (142.251.236.105)  25.890 ms
   209.85.142.97 (209.85.142.97)  19.741 ms
 9 209.85.242.78 (209.85.242.78)  22.319 ms
   209.85.241.70 (209.85.241.70)  20.953 ms
   108.170.228.8 (108.170.228.8)  21.144 ms
10 108.170.251.129 (108.170.251.129)  19.176 ms
   108.170.252.1 (108.170.252.1)  22.245 ms  18.352 ms
11 66.249.95.169 (66.249.95.169)  23.622 ms
   66.249.94.245 (66.249.94.245)  21.170 ms  21.789 ms
12 fra15s46-in-f3.1e100.net (172.217.16.131)  19.239 ms  19.512 ms  23.712 ms
MBAirGL:~ gerdlienemann$
```

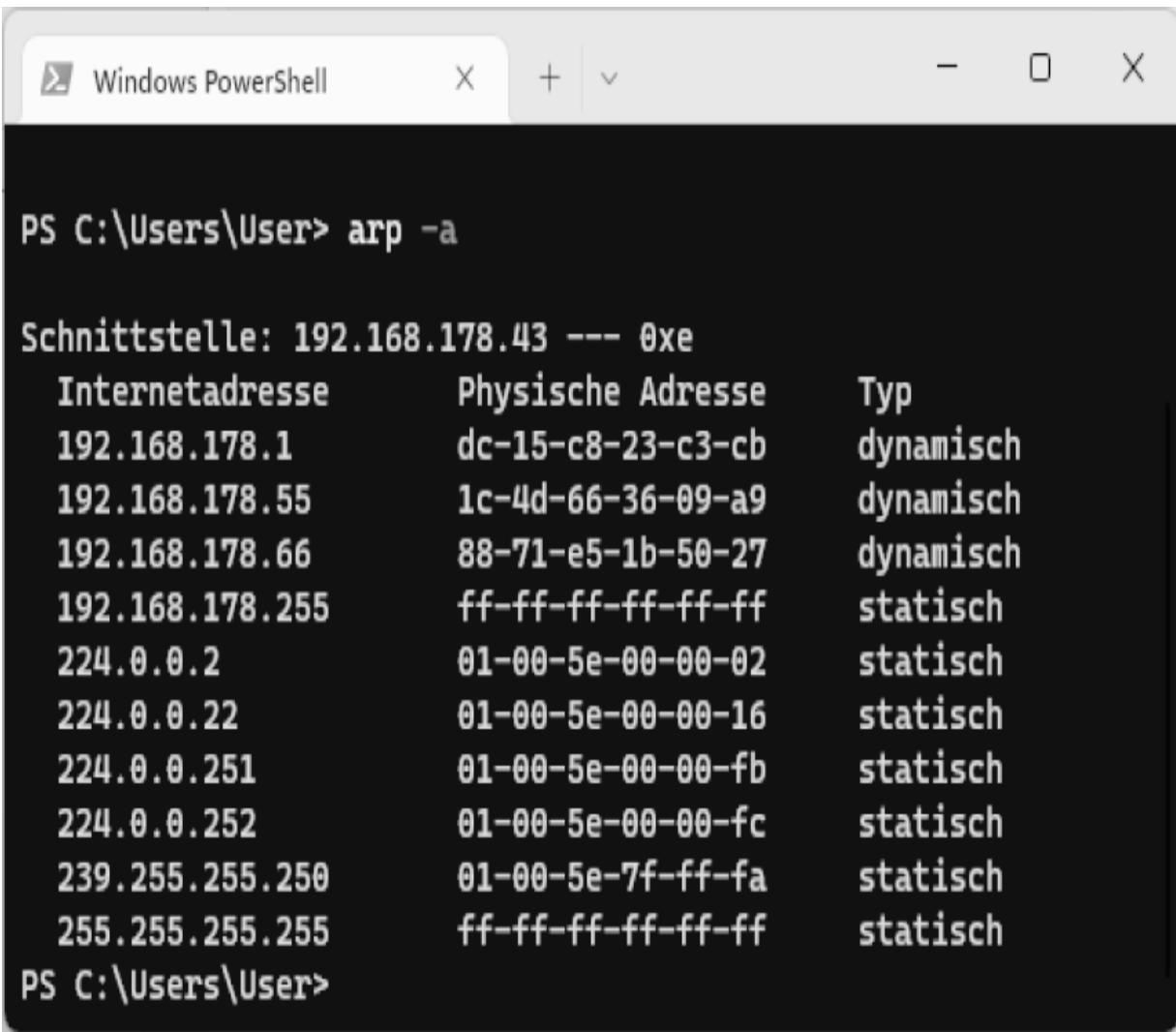
Abb. 8–16 *Beispiel einer Ablaufverfolgung mit TRACEROUTE*

Knotenadressen per ARP

In [Kapitel 2](#) dieses Buches wurde bereits erläutert, dass das Protokoll ARP (*Address Resolution Protocol*) dazu dient, einzelne Namen in ihre jeweilige MAC-Adresse (Knotenadresse) aufzulösen. Dies bedingt automatisch eine Darstellung der Adresse zu einem vorgegebenen Namen. Der TCP/IP-Protokollstack unterstützt diese Möglichkeit mit einer gleichlautenden Anweisung. Wird beispielsweise auf der Systemebene die folgende Anweisung

```
arp -a
```

eingegeben, so hat dies eine Anzeige der entsprechenden IP- und MAC-Adressen zur Folge. Das Ergebnis einer solchen Anweisung kann sich auf einem Windows-System beispielsweise wie in [Abbildung 8-17](#) darstellen.



```
PS C:\Users\User> arp -a

Schnittstelle: 192.168.178.43 --- 0xe
    Internetadresse    Physische Adresse    Typ
    -----
    192.168.178.1       dc-15-c8-23-c3-cb    dynamisch
    192.168.178.55      1c-4d-66-36-09-a9    dynamisch
    192.168.178.66      88-71-e5-1b-50-27    dynamisch
    192.168.178.255     ff-ff-ff-ff-ff-ff    statisch
    224.0.0.2           01-00-5e-00-00-02    statisch
    224.0.0.22          01-00-5e-00-00-16    statisch
    224.0.0.251         01-00-5e-00-00-fb    statisch
    224.0.0.252         01-00-5e-00-00-fc    statisch
    239.255.255.250     01-00-5e-7f-ff-fa    statisch
    255.255.255.255     ff-ff-ff-ff-ff-ff    statisch
PS C:\Users\User>
```

Abb. 8–17 Kommando »arp -a« – Anzeige zugehöriger Mac-Adressen

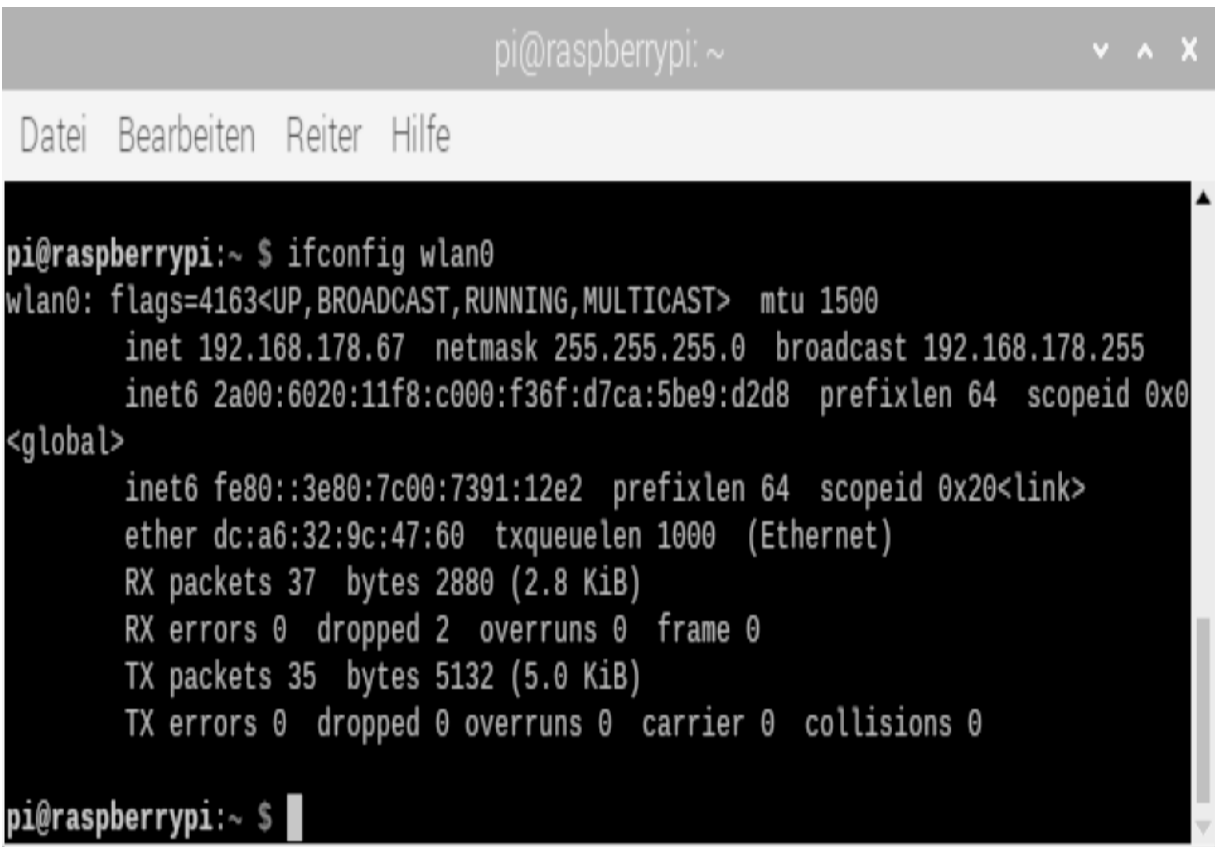
Aus dieser Aufstellung ist ersichtlich, mit welchen Hosts aktuell eine Verbindung besteht; zusätzlich wird zu jedem Host die zugehörige MAC-Adresse angezeigt.

Aktuelle Konfiguration

Mit der Anweisung IFCONFIG (Windows: IPCONFIG) können die verfügbaren (lokalen) Schnittstellen mit einem Blick überprüft werden. Um beispielsweise unter Debian Linux die erste WLAN-Schnittstelle (*wlan0*) zu überprüfen, muss folgende Anweisung eingesetzt werden:

```
ifconfig wlan0
```

Als mögliche Anzeige erscheint eine Darstellung wie in [Abbildung 8-18](#) ersichtlich.

A screenshot of a terminal window titled 'pi@raspberrypi: ~'. The window has a menu bar with 'Datei', 'Bearbeiten', 'Reiter', and 'Hilfe'. The terminal shows the command 'ifconfig wlan0' and its output. The output displays the configuration for the 'wlan0' interface, including flags, MTU, IP addresses (IPv4 and IPv6), netmask, broadcast address, prefix length, scope ID, and link type. It also shows statistics for RX and TX packets, bytes, errors, and dropped frames. The terminal ends with a prompt 'pi@raspberrypi:~ \$' and a cursor.

```
pi@raspberrypi:~ $ ifconfig wlan0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.178.67 netmask 255.255.255.0 broadcast 192.168.178.255
    inet6 2a00:6020:11f8:c000:f36f:d7ca:5be9:d2d8 prefixlen 64 scopeid 0x0
<global>
    inet6 fe80::3e80:7c00:7391:12e2 prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:9c:47:60 txqueuelen 1000 (Ethernet)
    RX packets 37 bytes 2880 (2.8 KiB)
    RX errors 0 dropped 2 overruns 0 frame 0
    TX packets 35 bytes 5132 (5.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

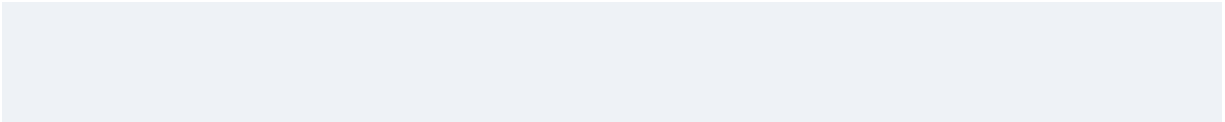
pi@raspberrypi:~ $
```

Abb. 8–18 Kommando »ifconfig wlan0« unter Debian Linux

Aus dieser Aufstellung ist erkennbar, dass die Schnittstelle *wlan0* aktiv ist (UP), und es werden die zugewiesenen Adressen und sonstigen Vorgaben (z.B. MTU für die Paketgröße) angezeigt.

Eine vergleichbare Anzeige kann unter Windows 11 nach Eingabe des Befehls

```
ipconfig /all
```



für den WLAN-Adapter eine Anzeige erzeugen, von der ein Auszug aus der bei Eingabe des Kommandos erzeugten Übersicht aller vorhandenen Netzwerk-Interfaces Abbildung 8-19 zeigt.

```
Windows PowerShell X + v - □ X

Drahtlos-LAN-Adapter WLAN:

Verbindungsspezifisches DNS-Suffix: fritz.box
Beschreibung. . . . . : Realtek 8821CU Wireless LAN 802.11ac USB NIC
Physische Adresse . . . . . : E0-E1-A9-31-B7-EA
DHCP aktiviert. . . . . : Ja
Autokonfiguration aktiviert . . . : Ja
IPv6-Adresse. . . . . : 2a00:6020:11f8:c000:c9b2:3a91:7686:ac38(Bevorzugt)
Temporäre IPv6-Adresse. . . . . : 2a00:6020:11f8:c000:3c6c:a407:ab94:6871(Bevorzugt)
Verbindungslokale IPv6-Adresse . . : fe80::c9b2:3a91:7686:ac38%14(Bevorzugt)
IPv4-Adresse . . . . . : 192.168.178.43(Bevorzugt)
Subnetzmaske . . . . . : 255.255.255.0
Lease erhalten. . . . . : Mittwoch, 6. Juli 2022 11:10:39
Lease läuft ab. . . . . : Mittwoch, 20. Juli 2022 19:23:04
Standardgateway . . . . . : fe80::de15:c8ff:fe23:c3cb%14
                          192.168.178.1
DHCP-Server . . . . . : 192.168.178.1
DHCPv6-IAID . . . . . : 182509993
DHCPv6-Client-DUID. . . . . : 00-01-00-01-29-D3-35-85-D8-BB-C1-98-01-4F
DNS-Server . . . . . : fd00::de15:c8ff:fe23:c3cb
                          192.168.178.1
NetBIOS über TCP/IP . . . . . : Aktiviert

Ethernet-Adapter Bluetooth-Netzwerkverbindung:
```

Abb. 8–19 Kommando »ipconfig /all« – Anzeige für den WLAN-Adapter

NSLOOKUP zur Nameserver-Suche

Die Anweisung NSLOOKUP dient der Fehlersuche bei DNS-Problemen, wenn also beispielsweise bestimmte Host-Namen nicht aufgelöst werden können oder nicht korrekt sind. Nach dem Aufruf von NSLOOKUP werden der Host-Name und die IP-Adresse des DNS-Servers angezeigt, der für das lokale System konfiguriert worden ist. Danach wird eine Eingabeaufforderung angezeigt. Mit der Eingabe eines Fragezeichens (?) werden die verfügbaren Befehle angezeigt. Um die IP-Adressen eines Hosts nachzusehen, der DNS verwendet, muss an der Eingabeaufforderung der Host-Name eingegeben und dies anschließend mit der Taste ENTER betätigt werden.

NSLOOKUP verwendet standardmäßig den DNS-Server, der für das entsprechende Endgerät (Rechner) konfiguriert worden ist, auf dem es ausgeführt wird. Es kann aber auch ein anderer DNS-Server verwendet werden, indem der Name des Servers nach der entsprechenden NSLOOKUP-Option angegeben wird.

Eine besonders hilfreiche Funktion zur Problembehandlung ist der Debug-Modus, der mit der NSLOOKUP-Anweisung *set debug* aktiviert werden kann. Im Debug-Modus listet NSLOOKUP sämtliche Schritte auf, die bei der Ausführung der

diversen Befehle ausgeführt werden. Eine typische Abfrage mittels NSLOOKUP (hier: dpunkt.de) kann sich beispielsweise wie folgt darstellen:

```
C:\Users\gerd>nslookup
```

```
Standardserver:  fritz.box
```

```
Address:  192.168.178.1
```

> set debug

```
> dpunkt.de
```

```
Server:  fritz.box
```

```
Address:  192.168.178.1
```

```
Got answer:
```

HEADER:

opcode = QUERY, id = 2, rcode = NXDOMAIN

header flags: response, auth. answer, want recursion, recursion avail.

questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:

dpunkt.de.fritz.box, type = A, class = IN

AUTHORITY RECORDS:

-> fritz.box

ttd = 9 (9 secs)

primary name server = fritz.box

responsible mail addr = admin.fritz.box

serial = 1657474363

refresh = 21600 (6 hours)

retry = 1800 (30 mins)

expire = 43200 (12 hours)

default TTL = 10 (10 secs)

Got answer:

HEADER:

opcode = QUERY, id = 3, rcode = NXDOMAIN

header flags: response, auth. answer, want recursion, recursion avail.

questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:

dpunkt.de.fritz.box, type = AAAA, class = IN

AUTHORITY RECORDS:

-> fritz.box

ttd = 9 (9 secs)

primary name server = fritz.box

responsible mail addr = admin.fritz.box

serial = 1657474363

refresh = 21600 (6 hours)

retry = 1800 (30 mins)

expire = 43200 (12 hours)

default TTL = 10 (10 secs)

Got answer:

HEADER:

opcode = QUERY, id = 4, rcode = NOERROR

header flags: response, want recursion, recursion avail.

questions = 1, answers = 1, authority records = 0, additional = 0

QUESTIONS:

dpunkt.de, type = A, class = IN

ANSWERS:

-> dpunkt.de

internet address = 188.40.189.211

ttl = 600 (10 mins)

Nicht autorisierende Antwort:

Got answer:

HEADER:

opcode = QUERY, id = 5, rcode = NOERROR

header flags: response, want recursion, recursion avail.

questions = 1, answers = 0, authority records = 1, additional = 0

QUESTIONS:

dpunkt.de, type = AAAA, class = IN

AUTHORITY RECORDS:

-> dpunkt.de

ttl = 600 (10 mins)

primary name server = ns1.xeneris.net

responsible mail addr = barabas.dpunkt.de

serial = 2022053103

refresh = 600 (10 mins)

retry = 600 (10 mins)

expire = 3600000 (41 days 16 hours)

default TTL = 600 (10 mins)

Name: dpunkt.de

Address: 188.40.189.211

>

Netzwerkanalyse mit WireShark

Die zuvor beschriebenen Analysetools können für die jeweiligen Betriebssysteme mit Bordmitteln verwendet werden. Separate Software ist in Fällen einfach zu behandelnder Fehlerszenarien nicht erforderlich. Bei komplexeren Problemen, bei denen man unter Umständen auch in einzelne Datenpakete schauen muss, lässt sich eine effektive Aussage meist nur nach Einsatz dedizierter Analysetools bewerkstelligen.

An dieser Stelle soll die Software »WireShark« gezeigt werden, da sie als ausgereifte Open-Source-Software mit zahlreichen Leistungsmerkmalen ausgestattet ist, aber noch sehr einfach zu bedienen ist. Bei der vorliegenden Version handelt es sich um die Version 3.6.6. Es erfolgt eine stetige

Weiterentwicklung, sodass sich bei Erscheinen des Buches diese Version vermutlich wieder überholt haben wird; für unser folgendes Beispielszenario ist dies jedoch irrelevant, da neuere Versionen auf die jeweiligen Vorläufer konsequent aufbauen. Eine damit verbundene angepasste Bedienung fällt i.d.R. sehr leicht.

Installation und Konfiguration

Die erforderlichen Dateien zur Installation erhält man auf der WireShark-Seite [wireshark.org](https://www.wireshark.org). Sie sind für die Plattformen Windows, macOS und den meisten Unix/Linux-Systemen verfügbar (unter Unix/Linux sind sie oft bereits Bestandteil der jeweiligen Distribution und müssen nur noch installiert werden). Siehe hierzu auch [Abbildung 8-20](#).



Abb. 8–20 Auswahl der 64-bit Version auf der [wireshark.org-Website](https://www.wireshark.org/Website)

Auf den Installationsvorgang soll hier im Detail nicht näher eingegangen werden. Allerdings darf nicht vergessen werden, den jeweils relevanten »Packet Capture«-(pcap-)Treiber (hier Npcap-Version 1.60) zu installieren, damit auf den verschiedenen Netzwerk-Schnittstellen des aufzeichnenden Rechners die Datenpakete mitgelesen werden können (siehe [Abb. 8–21](#)).

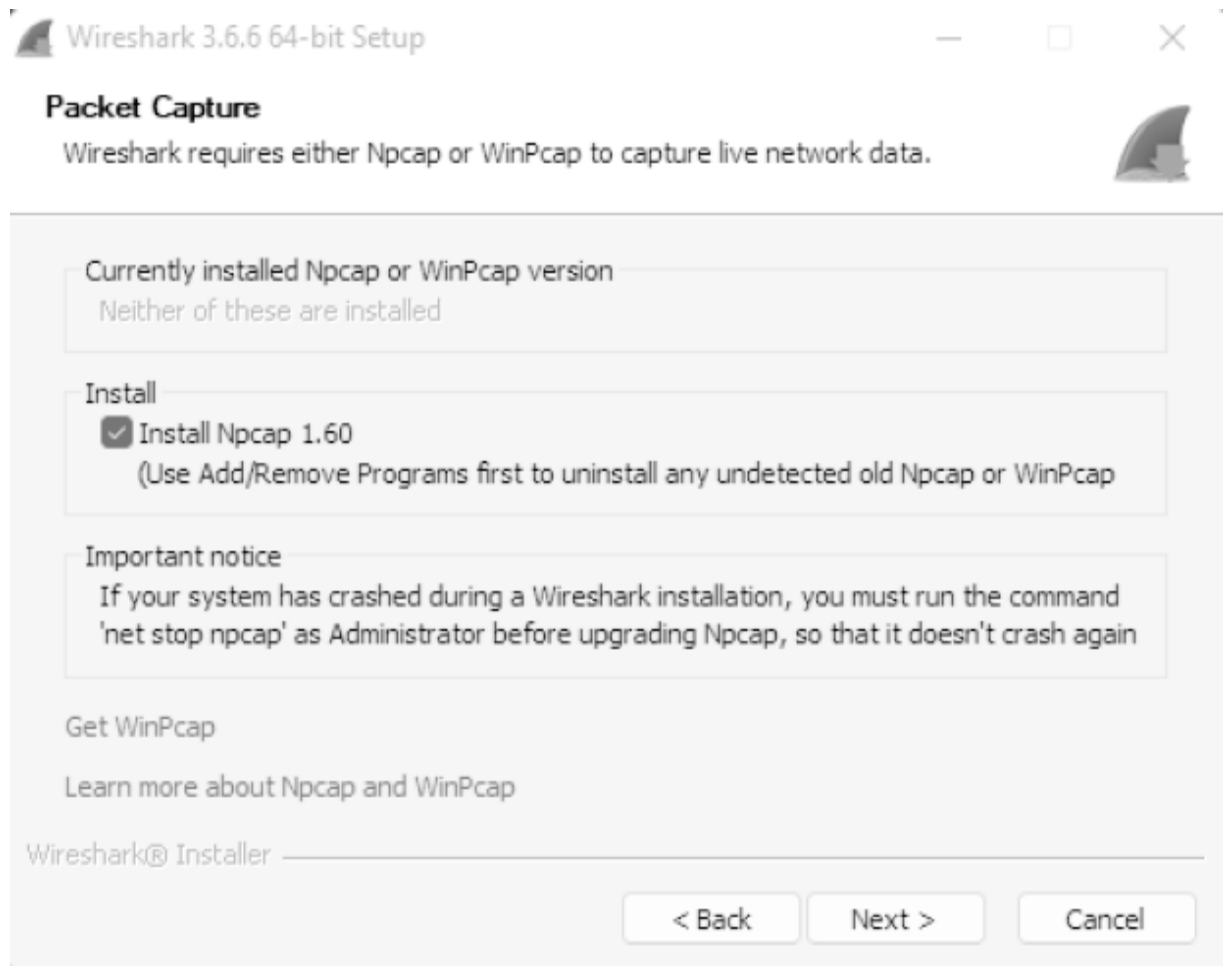


Abb. 8–21 *NPCAP 1.60 Treiber installieren*

Wird nun WireShark nach erfolgreicher Installation gestartet, erscheint der Startbildschirm wie in [Abbildung 8–22](#) gezeigt. Hier werden zunächst sämtliche erkannte Netzwerkschnittstellen angezeigt; inklusive ihrer aktuellen Aktivitäten als Liniendiagramm (hier lediglich auf der WLAN-Schnittstelle).



Willkommen zu Wireshark

Aufzeichnen

...mit diesem Filter:

Bluetooth-Netzwerkverbindung	_____
WLAN	_____
LAN-Verbindung* 2	_____
LAN-Verbindung* 1	_____
Adapter for loopback traffic capture	_____

Dokumentation

[Benutzerhandbuch \(en\)](#) · [Wiki \(en\)](#) · [Fragen und Antworten \(en\)](#) · [Mailing Listen \(en\)](#)

Sie nutzen Wireshark 3.6.6 (v3.6.6-0-g7d96674e2a30). Updates werden automatisch heruntergeladen.

Abb. 8–22 *WireShark-Startbildschirm – Anzeige aller Netzwerkadapter*

Soll nun für einen bestimmten Netzwerkadapter die Paketaufzeichnung gestartet werden, so markiert man diesen (in [Abb. 8–22](#) ist gerade der WLAN-Adapter markiert, da über diesen derzeit kommuniziert wird) und aktiviert die Aufzeichnung über den Menüpunkt »Aufzeichnen«.

Im folgenden Beispiel wird der Aufruf der Website »[dpunkt.de](#)« aufgezeichnet.

Szenario: Web-Surfing

Für unser geplantes sehr einfaches Beispiel wird einmal die Website »[dpunkt.de](#)« über den Internet-Browser aufgerufen. Das Ergebnis ist in [Abbildung 8–23](#) zu sehen.

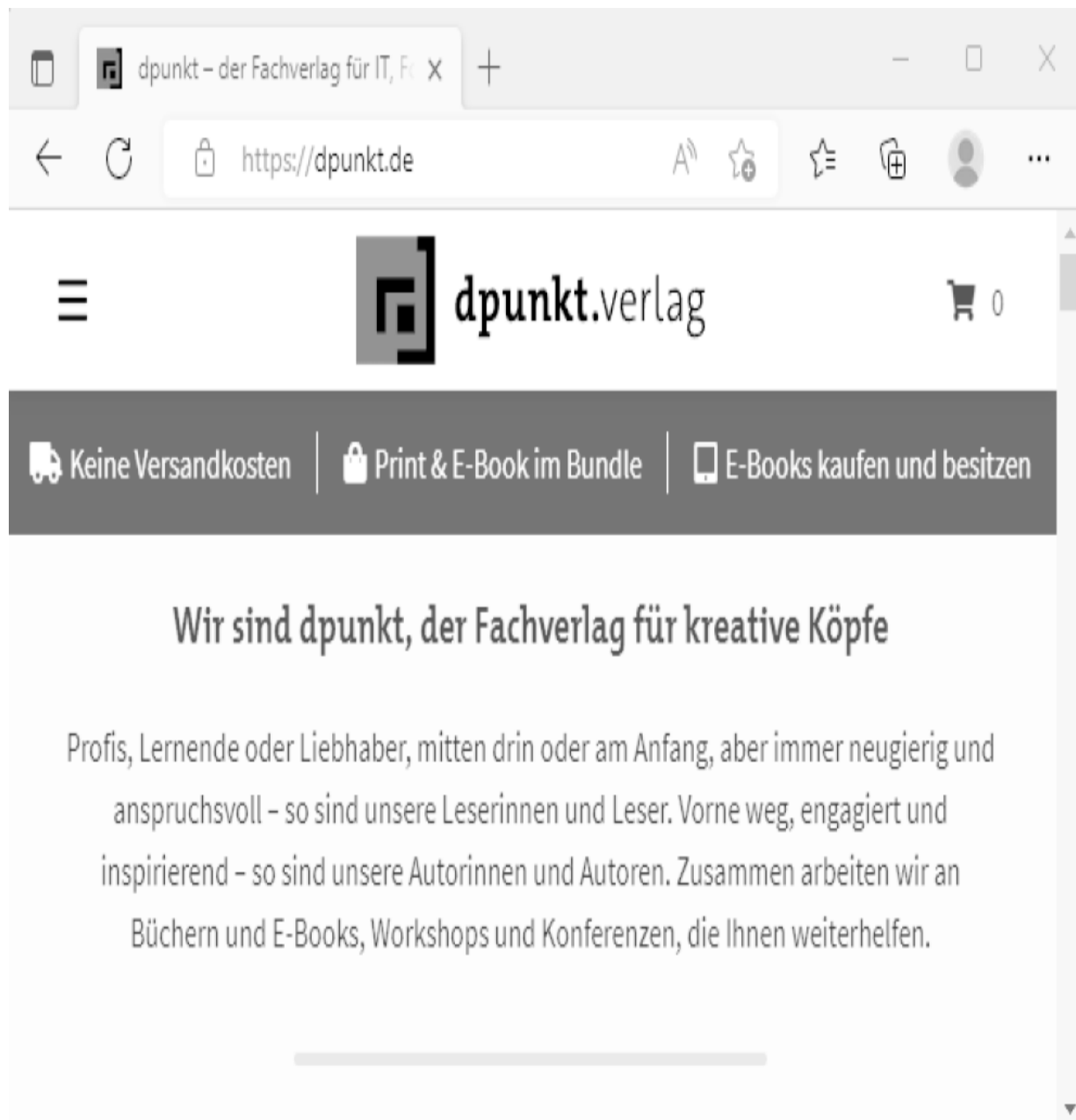


Abb. 8–23 Internet-Browser-Aufruf von »dpunkt.de«

In diesem Beispiel wird eine Paketaufzeichnung für lediglich 5 Sekunden vorgenommen. Wie aus [Abbildung 8–24](#) zu erkennen ist, wurden in diesem kurzen Zeitintervall bereits zahlreiche Datenpakete gelesen und zur Anzeige gebracht.

Über das »ping-Kommando« auf den Namen »[dpunkt.de](https://www.dpunkt.de)« wurde die IP-Adresse 188.40.189.211 ermittelt, die nun über den Anzeigefilter markiert wird, sodass die dann gefilterte Anzeige nur noch die relevanten Datenpakete enthält. Diese nunmehr fokussierte Darstellung ist [Abbildung 8–25](#) zu entnehmen.

Wireshark - *WLAN

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telefonie Wireless Tools Hilfe

ip.src == 188.40.189.211

No.	Time	Source	Destination	Protocol	Length
66	8.281136	188.40.189.211	192.168.178.43	TCP	66
71	8.300602	188.40.189.211	192.168.178.43	TCP	54
72	8.301978	188.40.189.211	192.168.178.43	TLSv1.3	304
79	8.320138	188.40.189.211	192.168.178.43	TCP	60
80	8.321233	188.40.189.211	192.168.178.43	TLSv1.3	341
81	8.321233	188.40.189.211	192.168.178.43	TLSv1.3	591
92	8.335172	188.40.189.211	192.168.178.43	TCP	2974
94	8.335288	188.40.189.211	192.168.178.43	TCP	4434
96	8.336207	188.40.189.211	192.168.178.43	TCP	2974
98	8.336268	188.40.189.211	192.168.178.43	TCP	4434
100	8.338326	188.40.189.211	192.168.178.43	TCP	1514
101	8.338326	188.40.189.211	192.168.178.43	TLSv1.3	1514
104	8.357654	188.40.189.211	192.168.178.43	TCP	1514
105	8.357654	188.40.189.211	192.168.178.43	TCP	1514

[Next Sequence Number: 251 (relative sequence number)]
 Acknowledgment Number: 611 (relative ack number)
 Acknowledgment number (raw): 1806730303

```

0000 e0 e1 a9 31 b7 ea dc 15 c8 23 c3 cb 08 00 45 00 ...1...#...E.
0010 01 22 91 ee 40 00 36 06 c5 17 bc 28 bd d3 c0 a8 ...@6... ( ...
0020 b2 2b 01 bb cb c9 02 a2 d9 90 6b b0 84 99 50 18 ...+...k...P.
0030 00 2a a8 87 00 00 16 03 03 00 80 02 00 00 7c 03 ...*...|...
0040 03 73 2d dc c5 f7 ee 9e 32 1b e9 97 03 a1 16 7e ...s...2...N
0050 15 98 eb 32 b8 9a fd 56 cc 62 4f a9 1b 2d 6a 88 ...2...V...b0...j.
0060 8e 20 a0 32 50 dd 6b 34 49 44 71 15 bb 32 c8 db ...2P.k4 IDq...2...
0070 2a d9 b2 7c 67 f8 6c 6c 09 b5 cd 25 b6 0b 7d 81 ...*...|g...ll...%...}...
0080 53 9d 13 02 00 00 34 00 2b 00 02 03 04 00 33 00 ...S...4...+...3...
0090 24 00 1d 00 20 3e 58 5f e5 f8 bb de 61 ab d8 52 ...$...>X...a...R
00a0 a3 1e 02 77 a7 63 78 a9 b5 33 3b 9c e6 61 16 47 ...w...cx...3...a...G
00b0 3d 7a d4 c4 11 00 29 00 02 00 00 14 03 03 00 01 ...=z...)...
00c0 01 17 03 03 00 20 29 0f f6 00 ab 89 27 1e 15 97 ...)...'...
00d0 32 9a c1 7a 22 58 69 38 9a b3 a0 ce 96 60 1a f9 ...2...z"Xi8...`...
00e0 be bf eb 14 a9 85 17 03 03 00 45 40 c3 ad 2f 3b ...E@.../;
  
```

Wireshark - wlanv2TP1.pcapng | Pakete: 240 | Angezeigt: 80 (33.3%) | Verworfen: 0 (0.0%) | Profil: Default

Abb. 8–25 Auf »[dpunkt.de](https://www.dpunkt.de)« bzw. 188.40.189.211 gefilterte Anzeige

Die Bildschirmanzeige ist dreigeteilt. Im oberen Drittel befindet sich die Datenpaket-Übersicht, im mittleren Teil findet sich die Darstellung der einzelnen Datenfelder eines Pakets und im unteren Teil der Inhalt des im mittleren Teil markierten Datenfelds.

Häufig werden Datenpakete aufgezeichnet, weil es innerhalb einer Kommunikation zu Verzögerungen kommt (Performance-Probleme) und ermittelt werden muss, welches Gerät diese Verzögerung verursacht. Dabei sind die Zeitstempel von entscheidender Bedeutung (in [Abbildung 8-25](#) sind das die Werte unter der Spalte »Time«). Werden die Datenpakete in chronologischer Reihenfolge betrachtet und es kommt von einem zum nächsten Datenpaket zu einer größeren Zeitdifferenz, so kann wesentlich gezielter nach der Ursache gesucht werden (beispielsweise wenn auf dem diese Zeitdifferenz verursachenden Rechner eine Datenbank läuft; nicht selten kommt es bei Datenbankzugriffen zu Verzögerungen, deren Ursache man dann gezielt analysieren kann).

Diverse Auswertungen

Einige Beispiele weiterer Auswertungen und Statistiken finden sich unter dem Menüpunkt »Statistiken« (siehe [Abb. 8-26](#) bis [8-](#)

28).

Details

Datei

Name: C:\Users\User\AppData\Local\Temp\wireshark_WLANW2STP1.pcapng
Länge: 187 kB
Hash (SHA256): c67f2d6696097ebafbdcf14c44b745935c40538b6202d74e02ba37b0e0176bec
Hash (RIPEMD 160): 9029981c1d29b75105cb5a349073269e38f69874
Hash (SHA1): 733601117c1c6ea84eb7902249cadd6ac665ceaf
Format: Wireshark/... - pcapng
Datenkapselung: Ethernet

Zeit

Erstes Paket: 2022-07-15 12:16:34
Letztes Paket: 2022-07-15 12:16:39
Zeitspanne: 00:00:05

Mitschnitt

Hardware: AMD Ryzen 5 5600G with Radeon Graphics (with SSE4.2)
BS: 64-bit Windows 10 (21H2), build 22000
Applikation: Dumpcap (Wireshark) 3.6.6 (v3.6.6-0-g7d96674e2a30)

Schnittstellen

<u>Schnittstelle</u>	<u>Verworfen Pakete</u>	<u>Mitschnittfilter</u>	<u>Linktyp</u>	<u>Paketgrößenlimit (snaplen)</u>
WLAN	0 (0.0%)	keine	Ethernet	262144 Byte

Statistik

<u>Messwerte</u>	<u>Aufgezeichnet</u>	<u>Angezeigt</u>	<u>Markiert</u>
Pakete	192	192 (100.0%)	—
Zeitspanne, s	5.018	5.018	—
Durchschnittliche pps	38.3	38.3	—
Durschnittliche Paketgröße, B	943	943	—
Byte	181027	181027 (100.0%)	0
Durschnittliche Byte/s	36 k	36 k	—
Durschnittliche Bit/s	288 k	288 k	—

Mitschnittdateikommentare

Aktualisieren

Kommentar speichern

Schließen

In die Zwischenablage kopieren

Hilfe

Abb. 8–26 *Allgemeine Übersicht zur durchgeführten
Paketaufzeichnung*

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ Destinations and Ports	159				0,0317	100%	1,3400	1,388
▼ 255.255.255.255	2				0,0004	1,26%	0,0200	0,000
▼ UDP	2				0,0004	100,00%	0,0200	0,000
161	2				0,0004	100,00%	0,0200	0,000
▼ 224.0.0.251	2				0,0004	1,26%	0,0100	4,718
▼ UDP	2				0,0004	100,00%	0,0100	4,718
5353	2				0,0004	100,00%	0,0100	4,718
▼ 213.95.190.5	6				0,0012	3,77%	0,0500	1,353
▼ TCP	6				0,0012	100,00%	0,0500	1,353
443	6				0,0012	100,00%	0,0500	1,353
▼ 204.79.197.219	3				0,0006	1,89%	0,0200	0,535
▼ TCP	3				0,0006	100,00%	0,0200	0,535
443	3				0,0006	100,00%	0,0200	0,535
▼ 204.79.197.203	2				0,0004	1,26%	0,0100	0,520
▼ TCP	2				0,0004	100,00%	0,0100	0,520
443	2				0,0004	100,00%	0,0100	0,520
▼ 20.234.93.27	1				0,0002	0,63%	0,0100	0,783
▼ TCP	1				0,0002	100,00%	0,0100	0,783
443	1				0,0002	100,00%	0,0100	0,783
▼ 192.168.178.43	108				0,0215	67,92%	0,9900	1,388
▼ TCP	108				0,0215	100,00%	0,9900	1,388
52385	10				0,0020	9,26%	0,1000	1,379
52368	91				0,0181	84,26%	0,9000	1,388
52361	1				0,0002	0,93%	0,0100	1,242
52359	1				0,0002	0,93%	0,0100	0,549
52358	1				0,0002	0,93%	0,0100	0,588
52357	1				0,0002	0,93%	0,0100	0,813
52356	1				0,0002	0,93%	0,0100	0,558
52348	1				0,0002	0,93%	0,0100	0,539
52347	1				0,0002	0,93%	0,0100	0,897
▼ 188.40.189.211	34				0,0068	21,38%	0,3200	1,394
▼ TCP	34				0,0068	100,00%	0,3200	1,394
443	34				0,0068	100,00%	0,3200	1,394
▼ 13.226.158.51	1				0,0002	0,63%	0,0100	0,535
▼ TCP	1				0,0002	100,00%	0,0100	0,535
443	1				0,0002	100,00%	0,0100	0,535

Anzeigefilter:

Anwenden

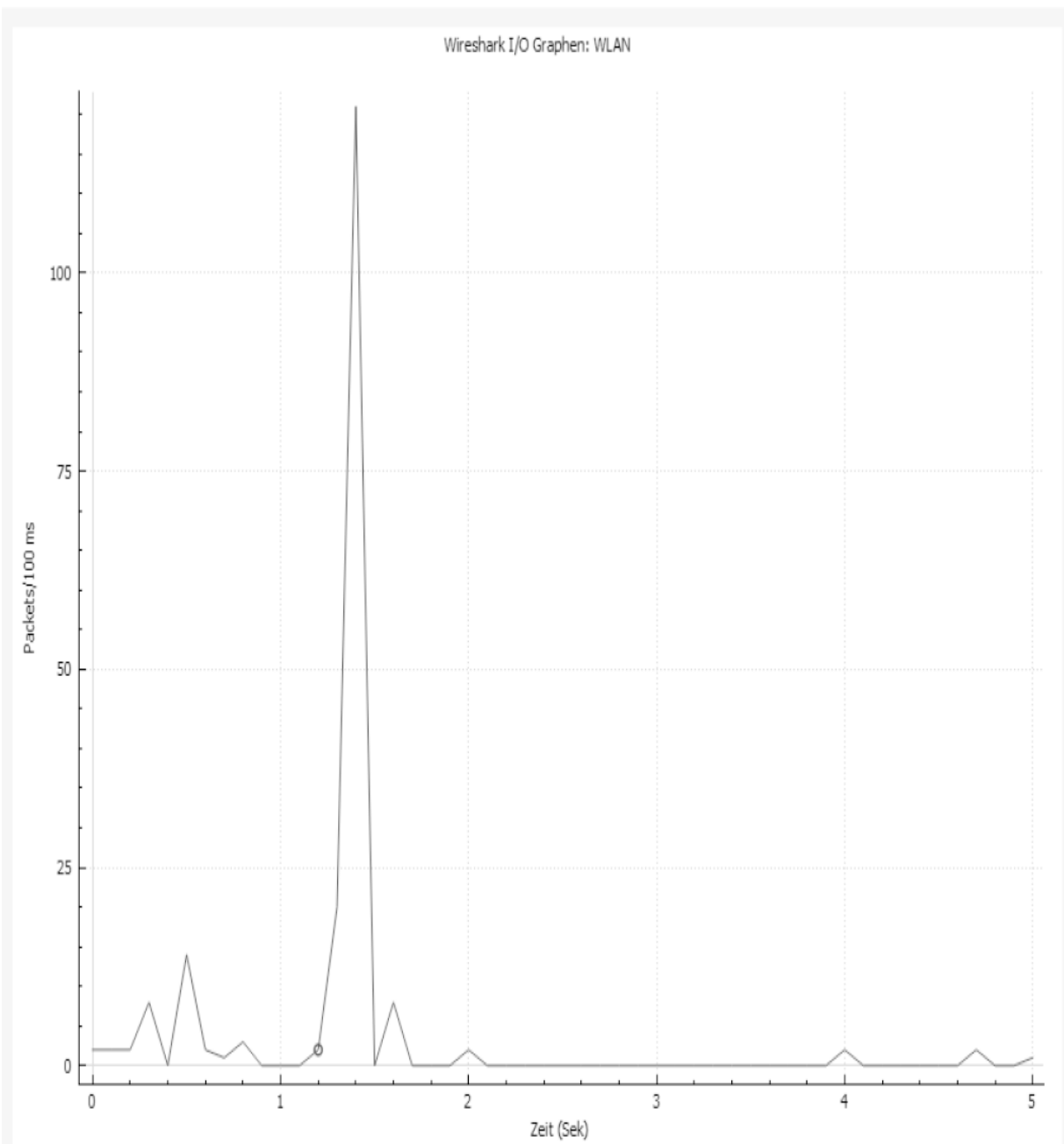
Kopieren

Speichern als...

Schließen

Abb. 8–27 *Anzahl Datenpakete pro Zieladresse – inkl. Protokoll-Ports*

Abbildung 8–27 liefert eine Übersicht der Datenpakete pro Zieladresse, einschließlich der Protokoll-Ports. Es ist beispielsweise erkennbar, dass beim Aufruf der »dpunkt.de«-Seite nicht der http-Standardport 80 verwendet wird, sondern Port 443, also die verschlüsselte Variante https.



Klicken zur Paketauswahl 36 (1.2s = 2).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period	Y Axis
☒	Alle Pakete		■	Line	Packets		None	1
☒	TCP Fehler	tcp.analysis.flags	■	Bar	Packets		None	1

+ - 📄 📊 Maus ☒ ziehen ☐ zoomen Intervall 100 ms ☐ Uhrzeit ☐ Logarithmische Skala ☒ Automatisch aktualisieren Zurücksetzen

Speichern als...

Kopieren

Kopieren von

Schließen

Hilfe

Abb. 8–28 *I/O-Graph mit 100-ms-Intervall über den 5-s-Aufzeichnungszeitraum*

Abbildung 8–28 zeigt einen Übersichtsgraphen der Input/Output-Situation der Datenpakete über den gesamten Aufzeichnungszeitraum von 5 Sekunden. Hier ist zwischen der ersten und zweiten Sekunde ein deutlicher Ausschlag zu erkennen. Dieser besagt, dass in diesem Zeitintervall eine hohe Anzahl von Datenpaketen übertragen wurde (>100). Ein Blick in die Aufzeichnungsübersicht der einzelnen Datenpakete bestätigt dies, indem die Zeitunterschiede zwischen den einzelnen Datenpaketen sehr klein sind.

Die in diesem Abschnitt beschriebenen Details zum Analysewerkzeug »Wire-Shark« sind natürlich nur sehr kleine Auszüge aus dem Gesamt-Funktionsumfang. Sie dienen lediglich der Veranschaulichung und der Vermittlung eines Eindrucks, wie umfangreich, aber zugleich sehr einfach Analysen durchgeführt werden können.

Anhang: Das neue TCP/IP-Umfeld

Nach den zurückliegenden Kapiteln folgt nun ein Abschnitt, der technische Protokolldetails nur am Rande thematisiert. Er versucht vielmehr, einen Schwerpunkt der über die letzten Jahre entstandenen neuen Umgebungen einer TCP/IP-basierten Kommunikation darzustellen:

Das »Internet der Dinge« (Internet of Things) IoT bzw. Industrie 4.0.

Es handelt sich dabei keineswegs um eine vollständige Betrachtung, sondern es soll lediglich exemplarisch dieser Schwerpunkt in groben Zügen vorgestellt werden.

Im Rahmen der Digitalisierung spielen IoT und die »Vierte Industrierevolution« (Industrie 4.0) eine besondere Rolle. Details hierzu folgen in den nächsten beiden Abschnitten.

Internet of Things (IoT)

Den Anfang dieser Betrachtung macht das Thema »Internet of Things« (IoT) oder im deutschen Sprachgebrauch auch »Internet der Dinge« genannt. Beginnt man mit dem Versuch einer Definition, so findet man auf der Homepage von Oracle (www.oracle.de) die folgenden Zeilen:

»Das Internet of Things (IoT) ist die Bezeichnung für das Netzwerk physischer Objekte (»Things«), die mit Sensoren, Software und anderer Technik ausgestattet sind, um diese mit anderen Geräten und Systemen über das Internet zu vernetzen, sodass zwischen den Objekten Daten ausgetauscht werden können. Diese Geräte reichen von normalen Haushaltsgegenständen bis hin zu anspruchsvollen Industriewerkzeugen.«

Diese einfache, aber sehr eingängige Definition liefert die Unterteilung in offenbar zwei Bereiche, in denen wir IoT-Technologien finden, deren Komponenten über TCP/IP kommunizieren:

- Industrie
- Privater Haushalt

Der industrielle Sektor umfasst allerdings auch Bereiche aus Verwaltung/Behörde, Gesundheitswesen oder auch militärische Bereiche, die hier aber nicht näher beschrieben werden.

IoT in der Industrie

Hier gilt der Grundsatz: kein IoT-Projekt ohne Business Case!

In einer Veröffentlichung der *Computerwoche Online* vom 26.08.2020 schrieb Raimund Schlotmann, dass es im Prinzip fünf Schritte zu einem erfolgreichen IoT-Geschäftsmodell gibt. Dabei sollte der Fokus nicht auf einer technologiebasierten Herangehensweise liegen, sondern es muss die Frage beantwortet werden: »Welchen Erfolg bringt IoT für das Unternehmen?« Hier eine Zusammenfassung seiner fünf Schritte:

- Digitaler Wirkungsmanager – Da ein IoT-Projekt stets Chefsache ist, sollte die Projektleitung direkt an die Geschäftsleitung berichten. Es ist ferner erforderlich, dass diese Instanz aus betriebswirtschaftlicher und kommunikativer Sicht Kunden wie Stakeholder einbindet. Seine Aufgabe ist u.a. die Definition der Wirkung von IoT auf die Kunden und das Unternehmen selbst.
- Geschäftsmodell-Nukleus – Die Einrichtung kleiner »Think Tanks« mit wenigen aber ausgesprochen kreativen Köpfen, die auch mal »um die Ecke denken können«, ist gefragt. Ausrichtung ist explizit nicht technisch, sondern geschäftsorientiert. Sonst ist der IoT-Gedanke nicht tragfähig. Am besten untermauert man die Relevanz durch einen kleinen schnell umsetzbaren, erfolgreichen Business Case.
- Überzeugende Anwendungsfälle – Um erfolgreich zu sein, reicht es nicht, eine überzeugende technische Machbarkeit

darzustellen, sondern es muss unzweifelhaft klar sein, wer für eine IoT-Lösung zahlen würde. Je offensichtlicher, desto besser. Es muss ein ausreichender »Pain Point« adressiert werden können.

- Härting per Stakeholder – Nachdem die Ideen für ein Projekt ausreichend entwickelt wurden, müssen substantielle Unterstützer für das Projekt motiviert werden, aktiv mitzuarbeiten. Ihr Status verändert sich vom »Beteiligten« zum »Betroffenen«. Dabei kann es sich sowohl um zentrale Funktionsträger im Unternehmen handeln als auch um Kunden.
- Erfolgsgrundlage schützen – Entscheidend bei der Umsetzung des Business Cases ist eine unbedingte Vermeidung von Aversion gegen die nach Projektabschluss anstehende »Veränderung«. Oft ist eine Vereinfachung des Projekts durch Trennung einzelner Disziplinen sinnvoll, wenn die eine Disziplin das Projekt zum Scheitern bringen kann.

IoT in der Industrie wird auch oft als IIoT (*Industrial Internet of Things*) bezeichnet. Ein oft genanntes Beispiel für den effektiven Einsatz im IIoT ist das »Predictive Maintenance«. Hier sorgen geschickt platzierte Sensoren in komplexen Maschinen für eine stetige Messung von Wartungszuständen, damit entweder automatische Reparaturprozesse in Gang

gesetzt werden können oder aber geeignetes Wartungspersonal informiert wird.

IoT im öffentlichen Sektor

Natürlich lassen sich auch für sämtliche öffentlichen Aufgaben (Staat, Land oder Kommune) IoT-Technologien einsetzen, die für eine erhöhte Effizienz und Kostenoptimierung sorgen können. So liefern Sensoren im Straßenbelag, in Gebäuden oder Brücken rechtzeitig Informationen über deren Zustand und können somit zu optimierten Reparaturprozessen beitragen.

Ein weites Feld für den öffentlichen Einsatz von IoT liegt im Verkehrsmanagement. Durch den Einsatz entsprechender Sensortechnik können Schadstoffe im Straßenverkehr minimiert werden, indem beispielsweise rechtzeitig über den fälligen Austausch von Partikelfiltern zentral informiert wird.

Ein anderes Beispiel liefern Kfz-Versicherer mit dem sogenannten »Telematik-Tarif«. Dabei werden dem Versicherten Rabatte angeboten, sofern er bereit ist, Fahrdaten (Bremsverhalten, Beschleunigungsverhalten) an die Versicherung zu übermitteln. Dies erfolgt über geeignete Sensoren im Fahrzeug und einer Online-Verbindung (zumeist über fest verbaute SIM-Karten). Die so zur Verfügung gestellten Daten ermöglichen es dem Versicherer dann bei einem

festgestellten umsichtigen Fahrverhalten des Fahrers, Entscheidungen zur Gewährung von Rabatten als »Belohnung« zu erteilen. Letztlich steht dahinter natürlich eine Reduktion des Risikos von für die Versicherung kostspieligen Schadensregulierungen.

IoT im privaten Haushalt

IoT-Technologien im privaten Haushalt sind heute bereits allgegenwärtig. In der einen oder anderen Weise haben wir alle schon mit dem Internet der Dinge Bekanntschaft gemacht. Hier einige Beispiele:

- Das Tracking beim Transportdienstleister, das oft per Mail oder Message-Dienst die minutengenaue Ankündigung der Lieferung einer bestellten Ware ankündigt
- Der automatisierte Service von Drucker-Patronenhändlern, die von Druckern Tintenstände regelmäßig online mitgeteilt bekommen und verzögerungsfrei Ersatz liefern
- Sensortechniken im »Smart Home«, die für das automatische Herunterfahren von Jalousien sorgen, sobald es dunkel wird, oder automatisch Bestellungen auslösen, wenn im Kühlschrank der Lebensmittelvorrat zur Neige geht
- Sprachsteuerung zahlreicher im »privaten Heimnetz« verbundenen Geräte (Licht, Entertainment, Heizung usw.)

Eine besondere Herausforderung im IoT-Umfeld besteht darin, die zigtausend Geräte mit einer gültigen IP-Adresse zu versorgen, damit diese auch in Netzwerke integriert werden können. Nicht zuletzt wegen eben dieser großen Anzahl von Komponenten nimmt die Adressierung der IPv6-Adressen zu (siehe hierzu auch [Abschnitt 3.4](#)), da deren Adressbereich um ein Vielfaches größer ist. Darüber hinaus kommt auf den Betreiber dieser Vielzahl von IP-Adressen auch ein anspruchsvolles Sicherheitsbedürfnis zu. In einem »Smart Home« können bei mittlerer Ausstattung mit Sensoren in verschiedenen Bereichen schon mal mehrere Hundert IP-Adressen zusammenkommen. Jede einzelne IP-Adresse birgt das Risiko einer Kompromittierung durch Hacker oder IT-affiner Bösewichte. Daher sollte auch die IP-Adresse einer außen am Haus angebrachten Überwachungskamera ausreichend abgesichert sein – entweder durch eine sehr gute Verschlüsselung bei Integration in ein Funknetz oder aber gleich mit einer sicheren Kabelverbindung.

Industrie 4.0

Fälschlicherweise werden die Begriffe »Industrie 4.0« und »Industrial Internet of Things« oft synonym verwendet. Dabei sollte Industrie 4.0 eher als Gesamtstrategie innerhalb der digitalen Transformation verstanden werden. Aber wie so oft

gibt es für derartige Schlagworte keine verbindliche Norm, sodass man sich mit einer vereinfachten Begrifflichkeit wohl abfinden muss.

In Abbildung A-1 wird der Versuch einer übersichtlichen Darstellung der bislang verwendeten Begriffe vorgenommen, der allerdings keinen Anspruch auf absolute Korrektheit oder Vollständigkeit erhebt. Er soll vielmehr auf einfache Weise das Verständnis der Zusammenhänge verdeutlichen.

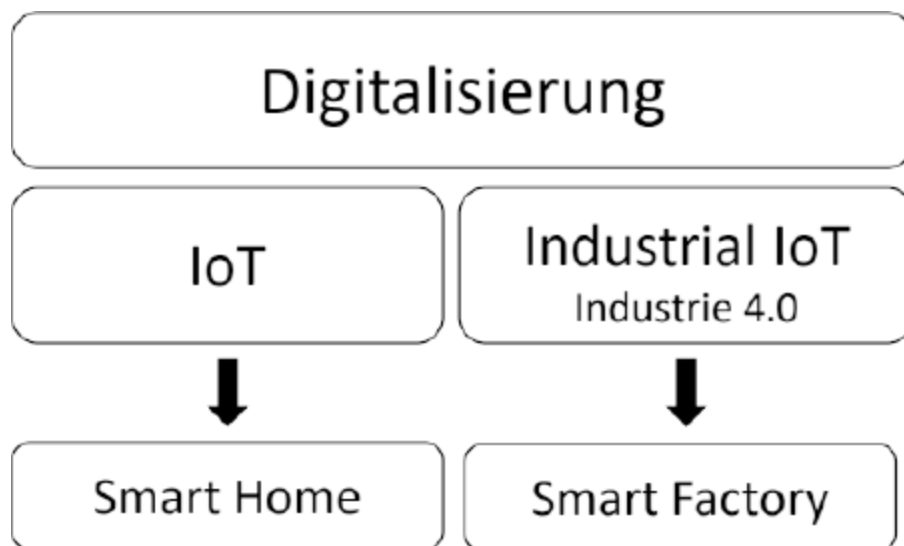


Abb. A-1 Versuch einer Darstellung IoT/IIoT und ihrer Einsatzbereiche

Die Begriffe »Smart Home« und »Smart Factory« stehen jeweils für die in den dargestellten Bereichen IoT und IIoT verwendeten Technologien wie beispielsweise Sensoren,

Vernetzungsart und -dichte, Maschine-zu-Maschine-Kommunikation, Robotik oder Service-Integration.

Wie bereits zuvor erwähnt, finden sich diese »Smart«-Technologien und -Lösungen auch in anderen Bereichen wie Smart Medicine, Smart Buildings, Smart City oder Smart Government, um nur einige Beispiele zu nennen. Die zentrale Gemeinsamkeit all dieser Technologien und Lösungen ist die Kommunikation (oft in Echtzeit) untereinander; und diese Kommunikation beruht im Wesentlichen stets auf den Protokollen der TCP/IP-Protokollfamilie.

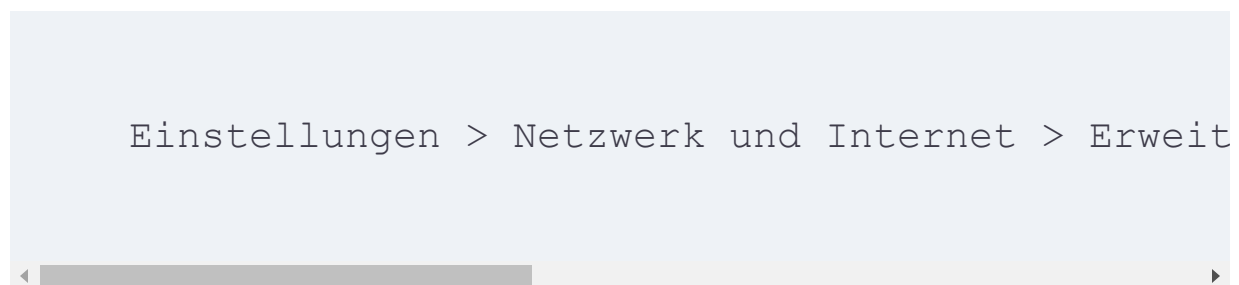
Anhang: TCP/IP-Konfigurationen

Wie die [Kapitel 1](#) bis [A](#) dieses Buches nahelegen, handelt es sich bei TCP/IP um eine Protokollfamilie, die weltweit einheitlich implementiert wurde: in Betriebssysteme, Anwendungssoftware oder auch Hardware-Komponenten. Allerdings sehen die Konfigurationen auf den einzelnen Plattformen oft sehr unterschiedlich aus, sodass eine Anpassung (sofern erforderlich) oft nicht einfach zu bewerkstelligen ist.

In den folgenden Abschnitten sollen die betriebssystemspezifischen Unterschiede dargestellt werden:

Microsoft Windows

Bei Microsoft Windows handelt es sich um die Version 11. Der Startpunkt für jede Art von Konfiguration sind die »Einstellungen« (das Zahnrad-Symbol auf der grafischen Oberfläche). Hier wählt man folgende Menü-Abschnitte:



Hier wurde »WLAN« deshalb ausgewählt, weil im betrachteten Szenario über die Funkschnittstelle konfiguriert werden soll. Soll die kabelgebundene Schnittstelle konfiguriert werden, ist der Menüpunkt »Ethernet« auszuwählen. Der Menüpunkt »Anzeigen zusätzlicher Eigenschaften« existiert für beide Netzwerke (siehe [Abb. B-1](#)).

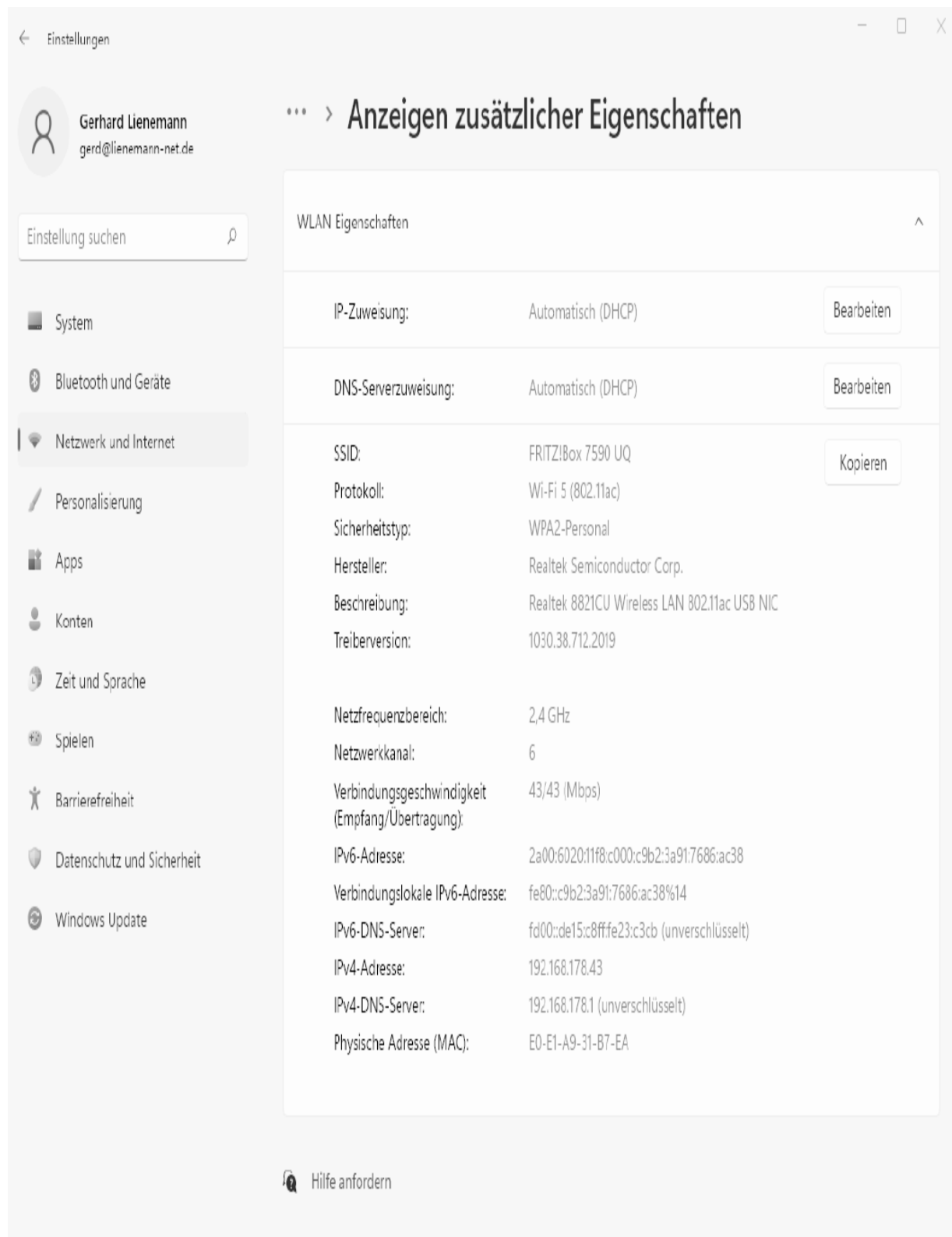


Abb. B–1 *WLAN > Anzeigen zusätzlicher Eigenschaften*

Nach Auswahl des zuletzt genannten Menüpunkts erlauben die dann aktivierbaren Schaltflächen »Bearbeiten« für »IP-Zuweisung« die Konfiguration einer automatischen IP-Adresszuordnung mit dem Eintrag »Automatisch (DHCP)« oder »Manuell« für eine manuelle IP-Adress-Konfiguration (siehe [Abb. B-2](#)).

Soll ein anderer DNS-Server zur Namensauflösung verwendet werden als der über DHCP zugeordnete, so kann dies über die DNS-Serverzuweisung erfolgen (siehe [Abb. B-3](#)).

IP-Einstellungen bearbeiten

Manuell



IPv4



Ein

IP-Adresse

Subnetzmaske

Gateway

Bevorzugter DNS

Bevorzugte DNS-Verschlüsselung

Nur unverschlüsselt



Alternativer DNS

Speichern

Abbrechen

Abb. B-2 *Manuelle Konfiguration der IP-Adressen*

DNS-Einstellungen bearbeiten

Manuell



IPv4



Ein

Bevorzugter DNS

Bevorzugte DNS-Verschlüsselung

Nur unverschlüsselt



Alternativer DNS

Alternative DNS-Verschlüsselung

Nur unverschlüsselt



IPv6



Aus

Speichern

Abbrechen

Abb. B–3 *Manuelle Konfiguration des DNS-Servers*

Die Netzwerk-Konfiguration unter Windows ermöglicht natürlich weit mehr Einstellungen (ähnlich bei den in den nächsten Abschnitten beschriebenen Betriebssystemen). Hier soll allerdings nur das Grundprinzip der Navigation, die Terminologie und die Funktionsweise der einzelnen Konfigurationsschritte und -menüs dargestellt werden.

Apple macOS

Beim macOS handelt es sich um die Version 12.4 (Monterey). Ältere Versionen sind allerdings sehr ähnlich zu konfigurieren. Hier findet man den Konfigurationsstart unter »Systemeinstellungen« und »Netzwerk«. [Abbildung B–4](#) zeigt das Einstiegsfenster.



Umgebung: Automatisch



WLAN

● Verbunden



HERO8 BLACK

● Nicht verbunden



Apple USB Adapter

● Nicht verbunden



Thunderbolt-Bridge

● Nicht verbunden



Thunderbolt-Bridge 2

● Nicht verbunden

Status: **Verbunden**

WLAN deaktivieren

„WLAN“ ist mit „FRITZ!Box 7590 UQ“ verbunden und hat die IP-Adresse 192.168.178.49.

Netzwerkname: FRITZ!Box 7590 UQ



- ☒ Automatisch mit diesem Netzwerk verbinden
- ☒ Zum Beitreten zu einem persönlichen Hotspot fragen
- ☒ Tracking der IP-Adresse beschränken

Du kannst das Tracking deiner IP-Adresse beschränken, indem die IP-Adresse vor bekannten Trackern in Mail und Safari verborgen wird.

- ☐ Auf neue Netzwerke hinweisen

Bekannte Netzwerke werden automatisch verbunden. Falls kein bekanntes Netzwerk vorhanden ist, musst du manuell ein Netzwerk auswählen.

☒ WLAN-Status in der Menüleiste anzeigen

Weitere Optionen ...



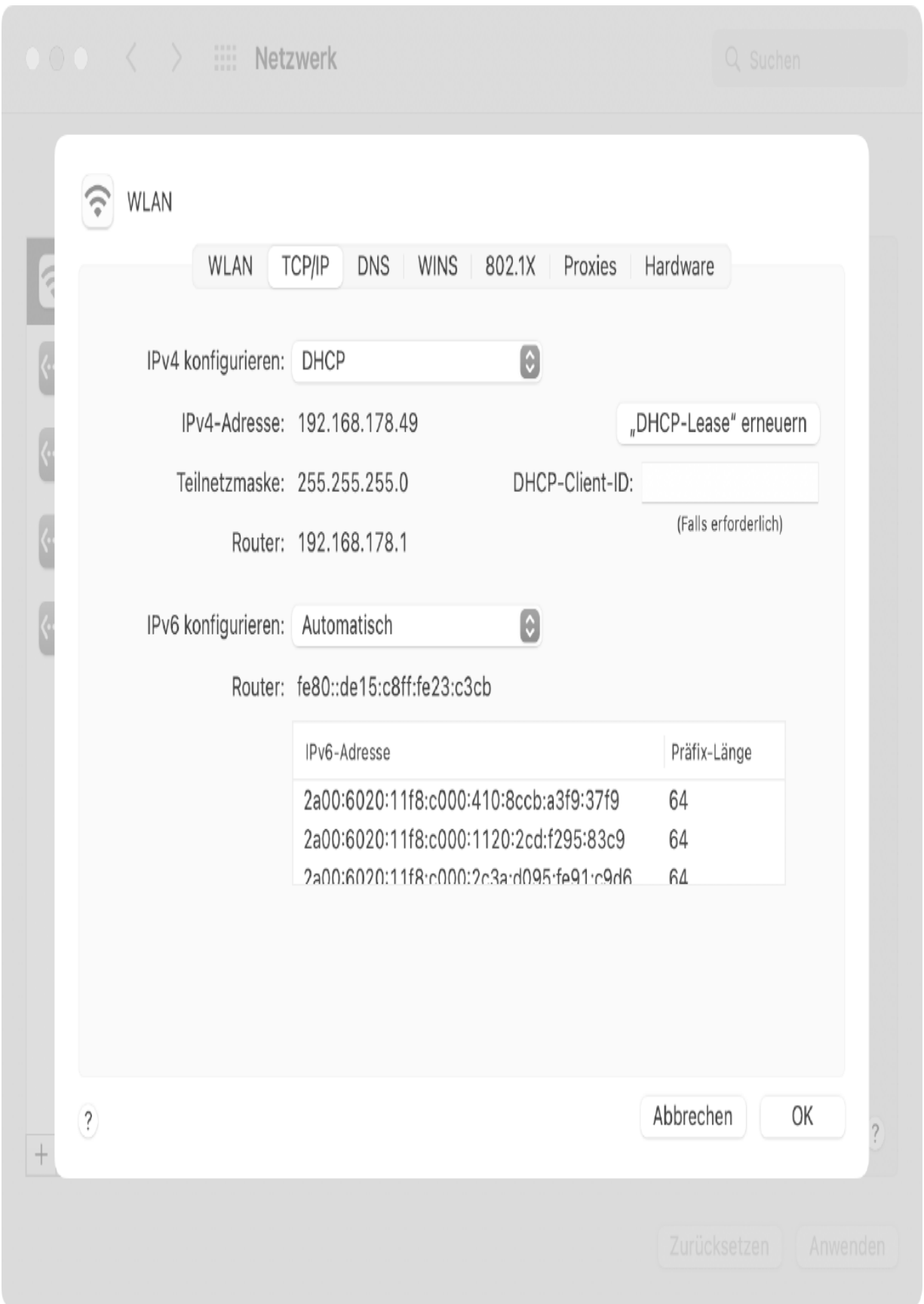
Zurücksetzen

Anwenden

Abb. B–4 *Einstieg Netzwerk-Konfiguration macOS 12.4*

Auf den ersten Blick erscheinen im linken Teil des Fensters die verfügbaren aktiven (grün) und nicht aktiven (rot) Netzwerk-Verbindungen. Hier ist lediglich die Funkverbindung »WLAN« aktiv. Das vorliegende Test-Netzwerk »Fritz!Box 7590 UQ« sollte natürlich im eigenen Heimnetzwerk aus Sicherheitsgründen auf einen anderen möglichst nicht nachvollziehbaren Namen geändert werden.

In den recht umfangreich konfigurierbaren Netzwerkbereich gelangt man über die Schaltfläche »Weitere Optionen« unten rechts in [Abbildung B–4](#).



WLAN

WLAN

TCP/IP

DNS

WINS

802.1X

Proxies

Hardware

IPv4 konfigurieren:

DHCP

IPv4-Adresse: 192.168.178.49

„DHCP-Lease“ erneuern

Teilnetzmaske: 255.255.255.0

DHCP-Client-ID:

(Falls erforderlich)

Router: 192.168.178.1

IPv6 konfigurieren:

Automatisch

Router: fe80::de15:c8ff:fe23:c3cb

IPv6-Adresse	Präfix-Länge
2a00:6020:11f8:c000:410:8ccb:a3f9:37f9	64
2a00:6020:11f8:c000:1120:2cd:f295:83c9	64
2a00:6020:11f8:c000:2c3a:d095:fe91:c9d6	64

Abbrechen

OK

Zurücksetzen

Anwenden

Abb. B-5 *macOS – Netzwerk-Konfiguration*

Abbildung B-5 zeigt die einzelnen Bereiche für die Netzwerk-Konfiguration. Hier ist das:

- WLAN
- TCP/IP
- DNS
- WINS
- 802.1X
- Proxies
- Hardware

Der sichtbare Bereich »TCP/IP« zeigt die IP-Adressen samt ihres Zuordnungsverfahrens (hier: DHCP und nicht manuell). Die IPv6-Adressen werden ebenfalls angezeigt.

Im Unterschied zu Windows kommt man hier grundsätzlich mit zwei Hierarchiestufen aus, die zudem recht übersichtlich über TABs angeordnet sind. Welches Prinzip man bevorzugt, ist allerdings Geschmacksache.

Debian Linux

Völlig anders sieht die Konfiguration von Netzwerkparametern unter Debian Linux aus. Üblicherweise erfolgt eine

Konfiguration stets manuell über entsprechende Einzelbefehle in einem System-Terminal. Es gibt allerdings eine sehr einfache grafische Möglichkeit, indem man das jeweilige Netzwerksymbol auf der Standard-GUI mit der rechten Maustaste anklickt (siehe [Abb. B-6](#)).

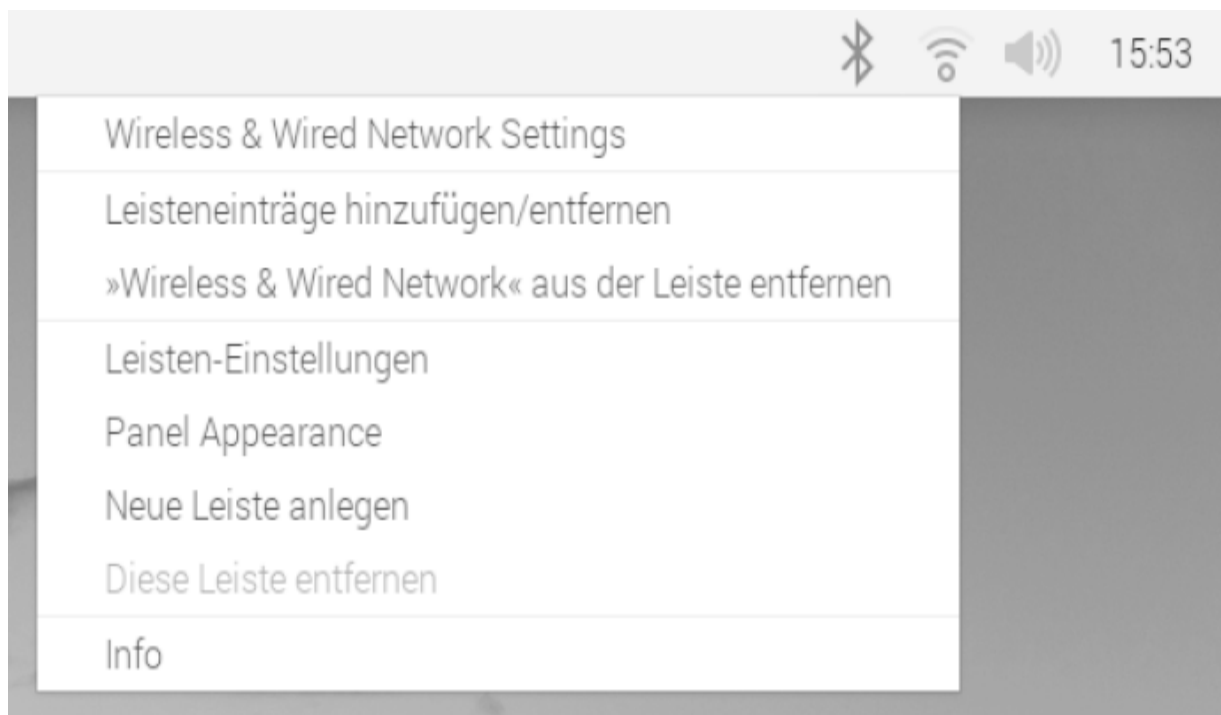


Abb. B-6 *Netzwerk-Konfiguration unter Debian Linux (GUI)*

In diesem Fall ist das der Menüpunkt »Wireless & Wired Network Settings«. Anschließend erhält man eine einfache Eingabemaske, in die IP-Adressen des eigenen Funk-Adapters, des Routers/Gateways und des DNS-Servers eingetragen werden können (siehe [Abb. B-7](#)).

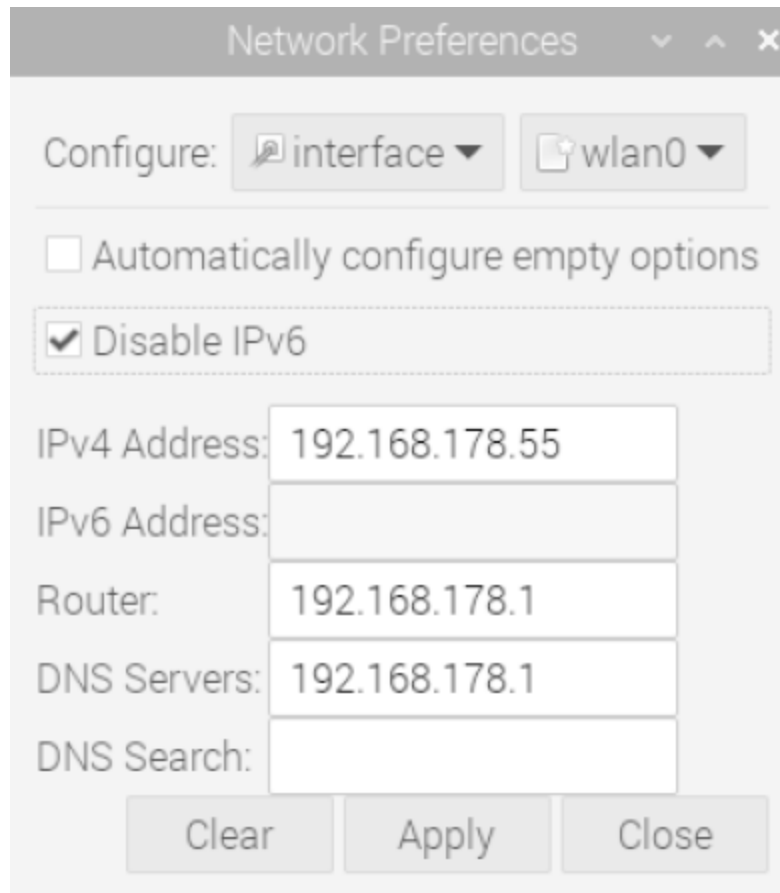


Abb. B-7 *Manuelles Eintragen von IP-Adressen unter Debian*

Wie bereits erwähnt, erfolgt die Netzwerk-Konfiguration (sofern sie manuell erfolgen soll und nicht Bestandteil einer automatischen auf DHCP basierten Einstellung ist) normalerweise mithilfe der Einzelkommandos (z.B. über das »ip«-Kommando). Es gibt natürlich auch ausgefeiltere grafische Oberflächen, die zur Konfiguration verwendet werden können. Sie hängen aber vom eingesetzten Linux-Betriebssystem ab und müssen dann oft nachträglich installiert werden.

Android

Auch wenn die Wahrscheinlichkeit nicht sehr groß ist, dass man die Netzwerkverbindungen eines Android-Smartphones oder -Tablets manuell konfiguriert (zumeist bleibt man bei der während der Initial-Installation geführten Konfiguration), so sollte man doch wissen, wo ggf. Anpassungen vorgenommen werden können. Dabei ist die Vorgehensweise stets ähnlich, auch wenn sich die Android-Systemversionen über die Jahre durch Updates verändern. Hat man das Symbol für die »Einstellungen« identifiziert (es handelt sich dabei immer um das Zahnrad-Symbol), so muss man in den darunter liegenden Menüpunkten lediglich den Bereich der »Verbindungen« finden und man gelangt – meist im »WLAN-Fenster« – zu den Netzwerkparametern, wie Abbildung B-8 zeigt.

16:55    •

   81% 

< FRITZ!Box 7590 UQ

Netzgeschwindigkeit

192 Mbps

Sicherheit

WPA2 PSK

IP-Adresse

192.168.178.23

Verwalten des Routers

Automatisch erneut verbinden



Erweitert



QR-Code



Entfernen



Abb. B–8 *WLAN-Konfiguration unter Android*

Neben der IP-Adresse können unter dem Menüpunkt »Erweitert« weitere Parameter konfiguriert werden (siehe [Abb. B–9](#)).

Erweitert

IP-Einstellungen

DHCP ▼

Proxy

Ohne ▼

Gebührenpflichtiges Netz

Automatisch erkennen

MAC-Adresstyp

Zufällige MAC

Abbrechen

Speichern



Abb. B–9 *Konfiguration weiterer Netzwerk-Parameter unter Android*

Hier lässt sich beispielsweise auch das IP-Adress-Zuordnungsverfahren festlegen (hier: DHCP; alternativ wären natürlich auch manuelle Einträge möglich).

Wie man sieht, erfolgt die Konfiguration eines Android-Smartphones normalerweise zwar weitgehend automatisch, manuelle Einstellungen sind aber ebenfalls möglich, sofern erforderlich. Dies ist beim IOS-System von Apple im nächsten Abschnitt auch nicht wesentlich anders.

Apple IOS

Beim Apple IOS (hier die Version 15.5) findet man die Netzwerkeinstellungen unter dem Menüpunkt »Einstellungen«. Die dann nach Auswahl des WLAN-Symbols aufgerufene Seite liefert bereits einige grundsätzliche Informationen (siehe [Abb. B–10](#)).

20:18



< Einstellungen

WLAN

WLAN



✓ FRITZ!Box 7590 UQ



NETZWERKE

FRITZ!Box 7490



Anderes ...

Auf Netze hinweisen Benachrichtigen >

Bekannte Netzwerke werden automatisch verbunden. Falls kein bekanntes Netzwerk vorhanden ist, wirst du auf verfügbare Netze hingewiesen.

Autom. mit Hotspot verbinden Hinw... >

Erlaube diesem Gerät, automatisch persönliche Hotspots in der Nähe zu erkennen, wenn kein WLAN verfügbar ist.

Abb. B-10 *Die »Einstellungen« unter Apple IOS 15.5*

Bei Auswahl des markierten WLAN-Netzwerks »FRITZ!Box 7590 UQ« gelangt man zu den Netzwerkparametern, wie [Abb. B-11](#) (Auszug der Seite über die Netzwerkparameter) zeigt.

20:19



< WLAN

FRITZ!Box 7590 UQ

Tracking der IP-Adresse
beschränken



Du kannst das Tracking deiner IP-Adresse beschränken, indem die IP-Adresse vor bekannten Trackern in Mail und Safari verborgen wird.

IPV4-ADRESSE

IP konfigurieren	Automatisch >
IP-Adresse	192.168.178.69
Teilnetzmaske	255.255.255.0
Router	192.168.178.1

IPV6-ADRESSE

IP-Adresse	2 Adressen >
Router	fe80::de15:c8ff:fe23:c3cb

DNS

DNS konfigurieren	Automatisch >
-------------------	---------------

HTTP-PROXY

Proxy konfigurieren	Aus >
---------------------	-------

Abb. B–11 *Netzwerkparameter unter Apple IOS 15.5*

Unter der Rubrik »IPv4-Adresse« findet sich erneut das Verfahren zur IP-Adresszuordnung. Hier steht »Automatisch« statt »DHCP«. Gemeint ist aber dasselbe Verfahren. Bei manueller Konfiguration können IP-Adressangaben sowie der Eintrag des Routers als Standard-Gateway dediziert vorgenommen werden. Ebenso besteht diese Möglichkeit bei der Auswahl des DNS-Dienstes.

Index

Ziffern

10 Gbit [5](#)

100VG-AnyLAN [11](#)

802.3an [5](#)

A

Abstract Syntax Notation One [240](#)

Access Control [21](#)

Access Control List [133](#), [260](#)

Access Point [15](#)

Accounting [157](#)

ACK [71](#)

ACK-Flag [276](#)

ACL [133](#)

Active Server Pages [204](#)

ActiveX [276](#)

Adaptive Routing [127](#)

Address Resolution Protocol [61](#), [307](#), [316](#)

Ad-hoc-Modus [14](#)

Administration Domain [107](#), [116](#)

Administration Domain Identifier [107](#)

Adressaufbau [82](#)

Adressierung [3](#)

Adresskonzept [79](#)

Adressregistrierung [81](#)

Adressumwandlung [115](#)

Adressvergabe

logisch [79](#)

physisch [79](#)

ADS [179](#)

Domäne [179](#)

ADSL2 [23](#)

ADSL2+ [23](#)

Advanced Encryption Standard [19](#)

AES [19](#)

Aggressive Exchange [290](#)

Aging [27](#)

AH [286](#)

American Registry for Internet Numbers [91](#)

Analyse-Tool [299](#)

Angriffsgefahr [271](#)

Anonymous FTP [201](#)

Anti-Replay-Service [283](#)

Anti-Spoofing-Mechanismus [276](#)

Anwendungsschicht [3](#)

Anzahl von Adressen [102](#)

Area Border Router [139](#)

ARP [61](#), [316](#)

ARP-Broadcast [307](#)

Asia-Pacific Network Information Center [91](#)

ASN.1 [240](#), [244](#)

ASN/1 [304](#)

ASP [204](#)

ATM [24](#)

Auditing [157](#)

Auflösung von Rechnernamen [170](#)

Authentication Header Protocol [286](#)

Authentication Header. Siehe AH [286](#)

Authentication Only Exchange [290](#)

Authentifizierung [219](#), [246](#), [283](#)

Authentisierung [111](#)

Autonome System [137](#)

B

Backbone-Area [137](#)

Bandbreite [299](#)

Base Exchange [290](#)

Berkley-Kommando [263](#)

Betriebssystem, Schwachstellen [280](#)

Betriebssystemanalyse [274](#)

Bewegtbilder [106](#)

BGP [153](#)

Bildschirmdarstellung [3](#)

Bluetooth [7](#), [19](#)

Body [213](#)

BootP [92](#), [155](#)

Bootstrap Protocol [92](#)

BPDU [28](#)

Breitbandtechnik [7](#)

Bridge [4](#)

Bridge Protocol Data Units [28](#)

Bridging [27](#), [122](#)

Broadband Technical [6](#)

Broadcast [92](#), [106](#), [124](#), [132](#)

Broadcast Domains [32](#)

Broadcast Network [139](#)

Broadcast-Adresse [41](#), [87](#)

Brücke [25](#)

BSD-UNIX [117](#)

C

Cable Television [7](#)

CATV [7](#)

CIDR [84](#), [95](#), [115](#)

Classless Inter-Domain Routing [95](#), [115](#)

CLNP [108](#)

Cluster [106](#)

CMIP [236](#)

CMIS [236](#)

CMOT [236](#)

Common Management Information Protocol [236](#)

Common Management Information Services [236](#)

Common Management Information Services Over TCP/IP [236](#)

Community-String [247](#)

Connection oriented [39](#)

Connectionless oriented [39](#)

Content Security System [294](#), [295](#)

Cracker [271](#), [272](#)

Criminal hacker [272](#)

CSMA/CA [16](#)

CSMA/CD [3](#), [4](#), [6](#), [8](#)

D

Data Encryption System [248](#)

Data Leakage [269](#)

Dateneingabe [3](#)

Datenintegrität [283](#)

Datenpaket [2](#)

Datensicherheit [254](#)

Datensicherungsschicht

Protokoll [40](#)

Datenstrom [111](#)

DCF [16](#)

DDNS [177](#)

DECnet [37](#)

Default Routing [125](#)

Demand Priority [6](#)

Demultiplexing [3](#)

Denial of Service Attack [274](#), [283](#)

DENIC [85](#), [90](#)

DES [110](#), [248](#), [249](#)

Designated Router [140](#)

DHCP [18](#), [94](#), [111](#)

DHCP-Server [110](#)

Diagnose und Fehlersuche [299](#)

Dienstgüte [110](#)

Digitale Unterschrift [295](#)

Direct Routing [125](#)

Direct Sequence Spread Spectrum [16](#)

Diskless Workstation [92](#)

Distributed Coordination Function [16](#)

Distributed Denial of Service Attack [275](#)

Distributed Queue Dual Bus [6](#)

DNS [111](#), [162](#), [168](#)

Dynamic [177](#)

Message Format [176](#)

Resolver [176](#)

Server [176](#)

DNS-Baum [172](#)

DNS-Datenbank [172](#)

DNS-Datenfluss [190](#)

DNS-Lookup [173](#)

DNS-Problem [318](#)

DNS-Query [278](#)

DNS-Response [279](#)

DNS-Ressourcendatei [185](#)

DNS-Sicherheitsproblem [277](#)

DNS-Spoofing [278](#)

DNS-Struktur [171](#)

DoI [290](#)

Domain Name Space [172](#)

Domain Name System [168](#), [169](#)

Domain of Interpretation [290](#)

Domäne [180](#)

Dotted Notation [80](#)

DQDB [6](#)

Drahtloses LAN [6](#)

DSAP [42](#)

DSL-Vectoring [24](#)

DSSS [16](#)

Dynamic DNS [177](#)

Dynamic Host Configuration Protocol [18](#), [94](#)

Dynamisches Routing [127](#)

E

Early Token Release [22](#)

E-Business [253](#)

ECHO REQUEST [59](#)

E-Commerce [102](#)

EIGRP [116](#)

Electronic Mail [212](#)

E-Mail [212](#)

Encapsulated Security Payload [282](#)

Protocol [287](#)

Ende-zu-Ende-Verbindung [115](#)

Ending Delimiter [21](#)

Enhanced IGRP [151](#)

ES-IS [152](#)

ESP [282](#), [286](#)

ESSID [14](#), [18](#)

Ethernet [3](#), [8](#)

Extended Service Set Identifier [14](#), [18](#)

External Data Representation [227](#)

External Gateway Protocol [120](#)

F

Fast Ethernet [4](#), [7](#), [10](#)

Fast IP [31](#)

FDDI [22](#)

FHSS [16](#)

Fiber Distributed Data Interface Network [22](#)

Fiber Optic [6](#)

File Transfer Protocol [195](#)

Final-Bit [43](#)

FIN-Flag [71](#)

Firewall [131](#)

Firewall-System [294](#)

Flatrate [23](#)

Flow Label [110](#), [111](#)

FQDN [172](#)

Fragmentierung [49](#)

Frei-Token [21](#)

Frequency Hopping Spread Spectrum [16](#)

FTP [195](#)

Anonymous [201](#)

Trivial [201](#)

Fully Qualified Host Name [80](#)

G

Gateway [34](#)

GET-Methode [208](#)

Gigabit [5](#)

Gigabit Ethernet [4](#), [7](#), [10](#)

Gigabit Media Independent Interface [11](#)

Glasfaserkabel [6](#)

GMII [10](#)

Grundschutzhandbuch [293](#)

GSM-Modem [156](#)

H

Hacker [271](#), [272](#)

Hardwarekomponente [25](#)

Hauptdomäne [172](#)

HDLC [42](#)

Header [213](#)

High Level Data Link Control [42](#)

High Speed Token Ring [8](#), [22](#)

Higher Level Interface Standard [4](#)

HLI [4](#)

Hop [129](#)

Hop-Anzahl [129](#)

Hop-Count [133](#)

Host-Datei [165](#)

Host-ID [85](#)

Host-Route [126](#)

Hosts (Datei) [165](#)

Hotspot [17](#)

HSTR [8](#)

HTML [202](#)

HTTP [202](#)

HTTP/NG [211](#)

HTTP-Message [204](#)

HTTP-Request [206](#)

HTTP-Response [207](#)

HyperText Markup Language [202](#)

HyperText Transfer Protocol [202](#)

I

IAB [81](#)

IANA [65](#)

ICMP [56](#), [131](#), [315](#)

ICMP-Header [277](#)

ICMP-Message [58](#)

Kategorien [56](#)

ICMP-Requests [273](#)

Identity Protection Exchange [290](#)

IDS/IRS-System [295](#)

IDS-Systeme [273](#)

IEEE [3](#)

IEEE 802.11 [6](#), [12](#)

IEEE 802.11n [13](#)

IEEE 802.3 [4](#)

IEEE 802.4 [5](#)

IEEE 802.5 [20](#)

IEEE 802.6 [6](#)

IEEE 802.8 [6](#)

IGRP [150](#), [151](#)

IKE [289](#)

IKE-Aggressive-Mode [291](#)

IKE-Main-Mode [291](#)

IKE-Modi [291](#)

IKMP [282](#)

IMAP [220](#)

Implementierung [117](#)

Indirect Routing [125](#)

Information Transfer Format [43](#)

Informational Exchange [290](#)

Infrastruktur-Modus [15](#)

Integrated Services LAN [6](#)

Internet Architecture Board [81](#)

Internet Assigned Numbers Authority [65](#)

Internet Control Message Protocol [56](#), [131](#), [307](#)

Internet der Dinge [329](#)

Internet Key Exchange [289](#)

Internet Key Management Protocol [282](#)

Internet Message Access Protocol 4 (IMAP4) [220](#)

Internet of Things [329](#)

Internet Protocol [46](#)

Internet Security Association and Key Management Protocol
[289](#)

Internetdomain [90](#)

Intra-Area Routing [137](#)

Intrusion Detection System [273](#), [294](#)

Intrusion Response System [273](#)

IoT [329](#)

IP [46](#), [58](#), [94](#)

IP Next Generation [106](#)

IP Security [282](#)

IP-Header [111](#)

IP-Multicast-Grouping-VLAN [33](#)

IPnG [106](#)

IPsec [19](#), [282](#)

IP-Segment [91](#)

IP-Spoofing [275](#)

IP-Switching [31](#)

IPv4

Adressen einkapseln [114](#)

IPv5 [104](#)

IPv6 [102](#), [104](#), [106](#)

Encapsulation [113](#)

IPv4-Adresse [114](#)

Kompatibilität [113](#)

Stand der Einführung [113](#)

IP-Version 6 [106](#)

IP-Versionen [105](#)

IRS-Systeme [273](#)

ISAKMP [282](#), [289](#)

IS-IS [152](#)

ISLAN [6](#)

J

Jam-Signal [9](#)

K

Kabelfernsehen [7](#)

Kabelnetz [7](#)

Kerberos [232](#)

Klasse-B-Adresse [103](#)

Knotenadresse [316](#)

L

LAN [1](#)

virtuell [32](#)

Lastverteilung [127](#)

Layer [2](#)

LDAP [179](#), [225](#)

Lichtwellenleiter [6](#)

Lightweight Directory Access Protocol [225](#)

LINK STATE ACKNOWLEDGEMENT [149](#)

LINK STATE REQUEST [148](#)

LINK STATE UPDATE [148](#)

Link-State-Update [140](#)

LLC [42](#)

LLC Protocol Data Unit [42](#)

Local Area Network [1](#)

Logical Link Control [42](#)

Logische Adressvergabe [79](#)

Loopback [307](#)

Loopback-Schnittstelle [307](#)

Loose Source Routing [52](#)

Low-Power-Wi-Fi [13](#)

LPDU [42](#)

LSA [140](#)

M

MAC [18](#), [41](#)

MAC-Adresse [316](#)

MAC-Grouping-VLAN [33](#)

MAN [6](#)

Management Information Base [237](#), [242](#)

Management Information Base II [303](#)

Maximum Transfer Unit [54](#), [120](#)

Mbit [6](#)

Media Access Control [18](#), [41](#)

Media Tap [306](#)

Metropolitan Area Network [6](#)

MIB [237](#), [240](#), [242](#), [246](#)

MIB II [303](#)

Migrationsphase [117](#)

MIME [204](#), [209](#), [214](#), [217](#)

Mirror-Port [306](#)

Modulare Exponentiation [291](#)

MTU [131](#), [318](#)

Multicast [123](#)

Multicast-Adresse [41](#), [83](#)

Multicast-Gruppenadresse [111](#)

Multicasting [111](#)

Multimedia [6](#)

Multiplexing [3](#), [64](#)

Multipurpose Internet Mail Extension [204](#), [209](#), [217](#)

MX-Record [169](#)

N

Namensauflösung [161](#), [171](#), [318](#)

Prinzip [162](#)

statisch [165](#)

Nameserver [172](#)

Nameserver-Lookup [173](#)

Namespace [179](#)

NAT [81](#), [89](#), [107](#), [115](#)

NETSTAT [311](#)

Network Address Translation [81](#), [89](#), [107](#), [115](#)

Network File System [227](#)

Network Information Centre [171](#)

Network Information Services [231](#)

Network Interface Controller [33](#)

Network Management Application [239](#)

Network Management Station [242](#)

Network Service Access Point [108](#)

Network-Grouping-VLAN [33](#)

Netz-ID [85](#)

Netz-Route [126](#)

Netzwerkadresse [2](#)

Netzwerkarchitektur [4](#)

Netzwerkeinsatz [2](#)

Netzwerkkomponente [306](#)

Netzwerkqualität [299](#)

Netzwerkschicht [2](#)

Protokolle [46](#)

Netzwerkstatistik [302](#)

Netzwerk-Trace [300](#)

Next Hop Resolution Protocol [31](#)

NFS [227](#)

NIC [33](#), [171](#)

NIS [231](#)

NMA [239](#)

NMS [242](#)

NSLOOKUP [174](#), [272](#), [318](#)

n-Standard [13](#)

O

OAKLEY [290](#)

Oakley [282](#)

Oktett [82](#)

Open Shortest Path First [137](#)

OSI [2](#)

OSPF [116](#), [129](#), [137](#)

Out-Of-Band Access [155](#)

P

Packet Accounting [157](#)

Party-MIB [249](#)

PCF [16](#)

PDU [246](#)

Penetration-Test [272](#)

Perfect Forward Secrecy [292](#)

Permanent Virtual Circuit [139](#)

Physical Layer [4](#)

Physikalische Verbindung [307](#)

Physische Adressvergabe [79](#)

Pinbelegung [2](#)

PING [307](#)

Ping of Death [277](#)

PKI [248](#), [295](#)

Plug and Play [111](#)

Point Coordination Function [16](#)

Poll-Bit [43](#)

Polling [6](#)

POP3 [218](#)

Popularität des World Wide Web [102](#)

Port [65](#)

Port-Grouping-VLAN [32](#)

Portnummer [31](#)

Port-Scanning [273](#)

Post Office Protocol Version 3 [218](#)

POST-Methode [208](#)

Pre-Shared Key [292](#)

Primary Name Server [165](#), [182](#)

Priorisierung [64](#)

Private-MIB [244](#)

Privater Adressbereich [88](#)

Protocol Data Unit [246](#)

Protokoll

Datensicherungsschicht [40](#)

Definition [40](#)

Funktionen [38](#)

Netzwerkschicht [46](#)

verbindungsloses [39](#)

verbindungsorientiert [39](#)

Proxy Agent [239](#)

Public Key Infrastructure [248](#), [295](#)

PVC [139](#)

Q

Qualitätsparameter [111](#)

Qualitätssicherung [110](#)

Quick-Mode [292](#)

R

RARP [63](#)

RAS [281](#)

Realm Specific Internet Protocol [116](#)

REDIRECT [58](#)

Referenzmodell [2](#), [4](#)

Remote Access Service [281](#)

Remote Network Monitoring [303](#)

Repeater [6](#), [25](#)

Request For Comment [1](#)

Resolver [171](#), [173](#)

Resource Records [188](#)

Ressourcendateien [175](#)

Retransmission [65](#), [73](#)

Reverse Address Resolution Protocol [63](#)

Reverse DNS-Lookup [174](#)

RFC [1](#)

RFCs 822 [213](#)

RIP [133](#)

R-Kommando [263](#)

RMON [303](#)

RMON-Probe [303](#)

ROAD [107](#)

ROAD-Arbeitsgruppe [107](#)

Root [172](#)

Root Domain [172](#)

Round Trip Time [73](#)

Router [2](#), [34](#), [58](#), [119](#)

Architektur [124](#)

Aufgaben [120](#)

Einsatzkriterien [130](#)

Router-Hop [315](#)

Router-Initialisierung [154](#)

Router-Steuerung [157](#)

Routing [5](#), [119](#)

adaptive [127](#)

Algorithmus [128](#)

dynamisches [127](#)

Protokolle [63](#)

statisches [126](#)

Verfahren [126](#)

ROuting and ADdressing [107](#)

Routing Discovery [152](#)

Routing Information Indicator [41](#)

Routing Information Protocol [133](#)

Routing-Information [311](#)

Routing-Problem [103](#)

Routing-Protokoll [132](#)

Routing-Tabelle [105](#), [115](#), [124](#)

RPC [227](#)

RSA [248](#)

RSIP [116](#)

RTT [73](#)

S

SAP [42](#)

Schichtenmodell [2](#)

SDLC [42](#)

Secondary Name Server [182](#)

Secure Socket Layer [281](#)

Security Parameter Index [287](#)

Security-Check [253](#)

Sequence Number [67](#), [287](#)

Service Access Point [44](#)

SGID [259](#)

SGMP [236](#)

Shared Media [29](#)

Sicherheit

externe [266](#)

interne [254](#)

organisatorische [269](#)

SILS [6](#)

Simple Gateway Monitoring Protocol [236](#)

Simple Internet Protocol Plus [107](#)

Simple Key Management Protocol [282](#)

Simple Mail Transfer Protocol [215](#)

Simple Network Management Protocol [39](#), [235](#)

Sitzungsschicht [3](#)

SKIP [282](#)

Skype [270](#)

Slow Convergence Problem [129](#)

SMI [237](#), [240](#)

SMTP [214](#)

SNA [37](#)

SNAP [42](#), [45](#)

SNMP [39](#), [235](#)

SNMP-Agent [239](#)

Socket [65](#)

Source [5](#)

SOURCE QUENCH [58](#)

Source Service Access Point [42](#)

Spanning Tree [26](#), [27](#)

Spezifikation [2](#)

Sprachdaten [106](#)

SSL-VPN [281](#)

Standard for Interoperable LAN Security [6](#)

Standard-UNIX-Rechte [256](#)

Starting Delimiter [21](#)

Statisches Routing [126](#)

Statusinformation [2](#)

Sticky Bit [258](#)

Store-and-Forward-Verfahren [30](#)

Subdomain [172](#)

Subnetwork Mask [85](#)

Subnetz [84](#), [90](#)

Subnetzmaske [85](#)

SUID [259](#)

Supernetting [105](#)

Supervisory Format [43](#)

SVC [139](#)

SVTX [258](#)

Switch [28](#)

Typen [31](#)

Switched Virtual Circuit [139](#)

Switching [122](#)

Synchronous Data Link Control [42](#)

SYN-Flag [70](#), [273](#)

SYN-Flooding [276](#)

T

Tag Switching [31](#)

TCP [66](#)

TCP and UDP with Bigger Addresses [108](#)

TCP/IP

Grundlagen [37](#)

TCP-Frame-Header [276](#)

TCP-Segment-Format [67](#)

TELNET [191](#)

TFTP [201](#)

Three Way Handshake [276](#)

TIME EXCEEDED [59](#)

Time To Live [50](#), [121](#), [189](#), [309](#), [315](#)

TLD [164](#), [171](#)

Token [21](#)

Token Bus [5](#)

Token Passing [20](#)

Token Ring [4](#), [5](#), [20](#)

Top Level Domain [164](#), [172](#)

Topologie [4](#)

TOS [137](#)

Trace [300](#)

TRACEROUTE [315](#)

TRACERT [315](#)

Transmission Control Protocol [66](#)

Transmission Control Protocol/Internet Protocol [37](#)

Transparent Bridging [27](#)

Transportmodus [285](#)

Transportschicht [3](#)

Trivial File Transfer Protocol [155](#), [201](#)

Trojaner [275](#)

TTL [50](#), [121](#), [176](#), [309](#), [315](#)

TUBA [108](#)

Tunnelmodus [286](#)

U

UCC [221](#)

UDP [39](#), [74](#)

Umgebungsvariable [209](#)

Unicast [122](#)

Unified Collaboration and Communication [221](#)

Uniform Resource Locator [203](#)

Universal Resource Identifier [206](#)

Universal/Local Address Bit [41](#)

UNIX-System [166](#)

URI [206](#)

URL [203](#)

User Datagram Protocol [39](#)

User Datagram Service [42](#)

V

Vectoring [24](#)

Verbindungsloses Protokoll [39](#)

Verbindungsorientiertes Protokoll [39](#)

Verbindungsschicht [2](#)

Verschlüsselung [295](#)

Vertraulichkeit [283](#)

Virtual Private Network [280](#)

Virtuelles LAN [32](#)

VLAN [5](#), [32](#)

IP-Multicast-Grouping [33](#)

MAC-Grouping [33](#)

Network-Grouping [33](#)

Port-Grouping [32](#)

Vollqualifizierter Domänenname [172](#)

Vollqualifizierter Name [80](#)

VPN [280](#)

W

WAN [1](#)

WEP [18](#)

Wide Area Network [1](#)

Wi-Fi HaLow [13](#)

Wifi Protected Access [19](#)

WIMAX [7](#)

Windowing [64](#), [69](#), [71](#)

Window-Size [72](#)

Wired Equivalent Privacy [18](#)

Wireless Gigabit [13](#)

Wireless Internet Service Provider [17](#)

Wireless Personal Area Network [7](#)

WireShark [321](#)

WISP [17](#)

WLAN [6](#)

Controller [15](#)

Hotspot [17](#)

Sicherheit [17](#)

Störquellen [17](#)

Zugriffsverfahren [16](#)

wlan0 [317](#)

WPA [19](#)

WPA2 [19](#)

WPAN [7](#), [19](#)

X

X.500 [226](#)

XDR [227](#), [230](#)

XID [43](#), [44](#)

Z

Zelle [14](#)

Zertifikat [295](#)

Zone [173](#)

Zugriffsverfahren [4](#), [6](#)



Rezensieren

Sie dieses Buch



Senden

Sie uns Ihre Rezension
unter www.dpunkt.de/rez



Erhalten

Sie Ihr Wunschbuch aus
unserem Verlagsangebot

